

601146

2 of 3

ESD-TDR- 63-474-11

MITRE SS-11

FIRST CONGRESS ON THE INFORMATION SYSTEM SCIENCES

SESSION 11

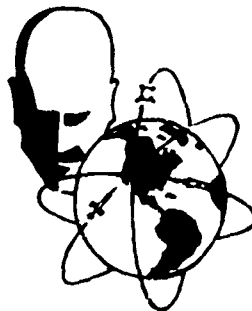
INFORMATION SYSTEM OPERATIONS ANALYSIS

TECHNICAL DOCUMENTARY REPORT NO. ESD-TDR- 63-474-11

MAY 1964

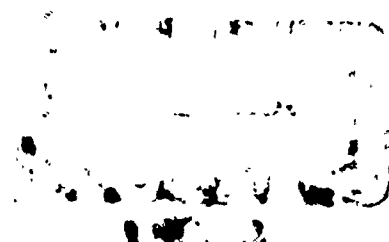
121-P # 2,75-

Prepared for
DIRECTORATE OF SYSTEM DESIGN
DEPUTY FOR TECHNOLOGY
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts



Project 704

Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts
Contract AF 33(600)-39852



Copies available at Office of Technical Services,
Department of Commerce.

Qualified requesters may obtain copies from DDC.
Orders will be expedited if placed through the librarian
or other person designated to request documents
from DDC.

When US Government drawings, specifications, or
other data are used for any purpose other than a
definitely related government procurement operation,
the government thereby incurs no responsibility
nor any obligation whatsoever; and the fact that the
government may have formulated, furnished, or in
any way supplied the said drawings, specifications,
or other data is not to be regarded by implication
or otherwise, as in any manner licensing the holder
or any other person or corporation, or conveying
any rights or permission to manufacture, use, or sell
any patented invention that may in any way be related
thereto.

Do not return this copy. Retain or destroy.

FIRST CONGRESS ON THE INFORMATION SYSTEM SCIENCES

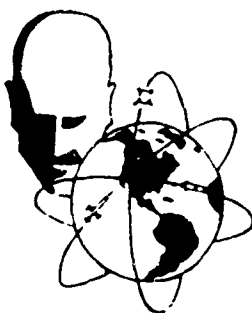
SESSION 11

INFORMATION SYSTEM OPERATIONS ANALYSIS

TECHNICAL DOCUMENTARY REPORT NO. ESD-TDR- 63-474-11

MAY 1964

Prepared for
DIRECTORATE OF SYSTEM DESIGN
DEPUTY FOR TECHNOLOGY
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts



Project 704

Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts
Contract AF 33(600)-39852

INFORMATION SYSTEM OPERATIONS ANALYSIS

Session Chairman: D. F. Votaw, Jr.

1. **INFORMATION SYSTEMS FOR MAN-MACHINE WAR GAMES**
John L. Donaldson and Joseph O. Harrison, Jr.
2. **FORMAL STRUCTURES FOR INFORMATION SYSTEM DESIGN**
Richard L. Van Horn
3. **OPERATIONS RESEARCH, INFORMATION TECHNOLOGY, AND INFORMATION SYSTEMS**
C. A. Wogrin and D. F. Votaw, Jr.

FIRST CONGRESS ON THE INFORMATION SYSTEM SCIENCES

Conducted at the Homestead, Hot Springs, Virginia

November 19-20, 1962

Chairman	Executive Manager for The MITRE Corporation
Edward Bennett	James Degan
Co-Chairman	Executive Manager for the Air Force Electronic Systems Division
Joseph Spiegel	Anthony Debons, Col., USAF

These preliminary manuscripts were distributed specifically for consideration by participants in the First Congress on the Information System Sciences. The Air Force Electronic Systems Division and The MITRE Corporation, as sponsors of the First Congress on the Information Systems Sciences, do not necessarily endorse the technical information or opinions expressed by the authors in these various working papers.

INFORMATION SYSTEMS FOR MAN-MACHINE WAR GAMES

John L. Donaldson and Joseph O. Harrison, Jr.

ABSTRACT

Two recent Research Analysis Corporation (RAC) activities in the development of techniques for man-machine war games are described. The first activity was the development of a semiautomatic war gaming system for THEATERSPIEL, one of the internal RAC war games. This system uses digital computer to perform the game assessment calculations, relieving the control group of this responsibility, and to serve as bookkeeper, providing complete numerical records, interval by interval, in a form suitable both for use during play and for postplay analysis. The system has been employed in a number of THEATERSPIEL plays. The second activity was the conduct of an experiment to test the feasibility of supporting Army war gaming by a remotely located digital computer. The experiment consisted of the assessment of the air operations and air defense portions of the 1961-1962 U. S. Army War College (USAWC) war games at Carlisle, Pa., by the RAC computer in Gaithersburg, Md. The experiment was successful both in demonstrating the feasibility of remote computational support and in improving the effectiveness of the USAWC war games.

FORMAL STRUCTURES FOR INFORMATION SYSTEM DESIGN

Richard L. Van Horn

ABSTRACT

Information system design has become a topic of prime importance. During this decade, the United States plans to spend billions on an information system venture known as "Command and Control." These electronic data systems will provide military commanders with information about our forces, the enemy, and nature. While specific hardware has been proposed to bolster present command and control structures, little has been done to design better information and decision systems. One step toward defining and solving some of the problems is to develop more formal structures. This involves formulation and investigation of alternatives, evaluation of cost and benefits associated with each alternative, and a mechanism for explicit communication of research.

Methods and examples are given for the evaluation of information and decision alternatives and for the analysis of information flow. System development is discussed, and formal programming structures, testing, and management factors are examined in detail. It is concluded that, although the application of formal analysis to information system problems has been limited in the past, the development of more formal structures for system design is possible and useful; that information system design and development, while still largely intuitive, can profit from a great deal more attention to formal techniques; and that, while the majority of technical people in the command and control field are specialists in hardware design, the major problems lie in determining information requirements, selecting good decision rules, and developing systems to implement these information structures and decision rules.

OPERATIONS RESEARCH, INFORMATION TECHNOLOGY,
AND INFORMATION SYSTEMS

C. A. Wogrin and D. F. Votaw, Jr.

ABSTRACT

The fundamental characteristics of information systems are described, and the important relationships between operations research and information technology are detailed. The discussion emphasizes that: (a) operations research can contribute significantly to the structuring of information systems and (b) the most significant contribution of operations research to information-system development can be made at the interface between the command (management) and the information system used by the command.

INFORMATION SYSTEMS FOR MAN-MACHINE WAR GAMES

FOREWORD

The authors are indebted to members of the Research Analysis Corporation (RAC) Computing Laboratory and the THEATERSPIEL study group for assistance, advice, and comments in connection with the development of the semiautomatic war gaming system. Special appreciation is expressed to Mr. George E. Clark, Jr., Manager, computing laboratory; Maj. Richard G. Williams, USA (Ret); Thomas R. Shaw; and Billy L. Himes, Sr.

The data-transmission study was the result of the combined efforts of the RAC, the U. S. Army War College and the U. S. Army Strategic Communications Command, with assistance from the Office of the Chief of Research and Development, Department of the Army; the U. S. Army Signal Communications Security Agency; the Office of the Chief Signal Officer, Department of the Army; the U. S. Army Security Agency; and the National Security Agency.

In connection with the data-transmission study, the authors wish to acknowledge the assistance and cooperation of Col. Jerry S. Addington, USA; USAWC; Capt. Douglas E. Price, USA, USASCC; Edward R. Fenske, USASCC; Charles W. Twigg, USASCSA; Col. LeRoy D. Brummitt, USA, OCRD, and RAC military advisor; George E. Clark, Jr. and Joseph B. Creegan, Jr., RAC; Capt. John Dorsett, USN (Ret), RAC; and Paul Michelsen, Harry L. Crow, and Everett W. Whitmarsh of RAC.

INFORMATION SYSTEMS FOR MAN-MACHINE WAR GAMES*

John L. Donaldson and Joseph O. Harrison, Jr. **

SECTION I

INTRODUCTION

An area in which the information system sciences are beginning to be useful is military war gaming. Through the years, military staffs have used war gaming for testing plans and for training personnel. Within the last decade, war gaming has been receiving increased emphasis as a tool of operations analysis.

The degree of formality with which information systems are applied to war games varies from one game to another. At one extreme is the completely mechanized war game or computer simulation. In Army war gaming, simulations usually have been confined to small unit actions whose inputs are tangible, quantitative, and measurable. Since human beings do not participate during the course of the play, the simulation can be handled completely on a computer. This results in very rapid execution, permitting repeated plays with large-scale variations of input conditions and chance factors. Simulations are capable of representing an action in sufficient detail to permit the inclusion of virtually every major attribute of a weapon system that bears on the outcome of an engagement.

*This paper is based in part on RAC-TP-59, "A Semiautomatic War Gaming System," June 1962, and RAC-TP-66, "A Data-Transmission Study," August 1962, FOR OFFICIAL USE ONLY. Also a portion of the material was presented previously at the Seventh Conference on the Design of Experiments in Army Research, Development, and Testing, October 18-20, 1961.

**The Research Analysis Corporation, Bethesda, Maryland.

At the other extreme is the completely manual war game. In comparison with computer simulations, manual war games are frequently inefficient. They are generally limited by the speed at which human beings can think and calculate. Hence, in a manual war game, there can be few repetitions of play and no wholesale variation of inputs and chance factors, as is done with simulations. However, manual war games can be used in situations where the rules for making decisions are not all specified in advance, some of the decisions being made instead on the basis of judgment by the game participants as the play proceeds. This gives the manual war game a somewhat broader range of applicability and more flexibility than the simulation. A manual war game provides an orderly method of combining the scientific knowledge and military judgment of experts in diverse fields; it ensures that a military situation will be considered from both sides – ours and the enemy's – and it capitalizes on the ingenuity of human participants to a great degree. Manual war games are also useful for training based on the element of human participation.

Recently, considerable effort has been devoted to developing a hybrid type of war game, combining to the greatest extent possible the advantages of both the computer simulation and the manual war game – the man-machine war game. In every war game, four functions must be performed: decision-making, computing, bookkeeping, and transmission of data, including display of results. Man-machine war games generally aim at mechanizing the last three of these functions. Rather elaborate physical equipment, including communications systems and display devices, is being developed to facilitate the communication between man and machine.

Man-machine war games will, of course, never achieve the extreme speeds which are obtained in pure computer simulations, since the mechanized operations must wait for human decisions. Consequently, not all the advantages of a pure computer simulation are possessed by the man-machine game.

However, an appreciable savings in time can be realized by eliminating some of the human delays. Moreover, this approach does provide for the benefits to be accrued from the inclusion of informed military judgment and experience. Such a war game also provides for more uniform and objective refereeing than a completely manual one does. It permits automatic recording of intermediate and final results, and frees human participants to concentrate on the substance of the game rather than on the mechanics of rules and formulas.

RAC has, for some years, been engaged in the development of man-machine war games of various types. This paper describes two of its recent activities in this field.

SECTION II

A SEMIAUTOMATIC WAR GAMING SYSTEM

GAME ENVIRONMENT

In the semiautomatic system discussed in this part of the paper, the digital computer fulfills two functions: first, it performs the game-assessment calculations, relieving the control group of this tedious, time-consuming responsibility, and second, it serves as a bookkeeper, providing complete numerical records of the play, interval by interval, in a form suitable for postgame analysis. To appreciate this application of a computer and its consequences, the reader must first be familiarized with the system within which the computer operates.

The war-gaming system can be considered as a sequence of related events, the relation being what might be termed an "information flow." Thus, for each event there is an input (which is the result of some prior event), some function that prescribes the manner in which this input is to be processed, and an output that is the result of this function (which will be input to the next event). By defining all events individually with regard to their inputs, functions, and outputs, the system as a whole is described. This section will examine the system in this manner, with one exception: the function of the computer operation, and its execution, will be the subject of a detailed discussion in the second section; in the present section, computer input-output will be discussed only to the extent necessary for the continuity of development.

The events occurring within the system can be divided into three phases: pregame planning, game play, and postgame analysis. Although each of these phases will be examined separately, it should be remembered that in reality they do not operate independently, since they, too are related by an information flow, or input-output process.

Pregame Planning

Once the study directive has been received and it has been decided that war gaming is an appropriate method of solution of the problem, the pregame planning phase is begun. The initial effort of this phase is to obtain a satisfactory statement of the problem together with specifications of the purpose and objectives of the game. This does not preclude the possibility that in the later stages of this phase it may be necessary to redefine the problem and objectives repeatedly; however, at the outset at least some general statement of purpose is a prerequisite to further development.

After the purpose has been determined, preparations for the game proceed along two parallel paths. Both the substantive and methodological aspects of play must be described. Consistent with the outlined objectives, the game environment must be established. This includes choosing a locale, developing a scenario, and collecting, organizing, and processing pertinent data. The choice of locale consists of selecting the geographical sector in which the game is to be played, of a size commensurate with the level of aggregation desired. The scenario includes the description of the political, economic, and cultural aspects of the environment leading up to the conflict. Also, a part of the scenario are the TOEs of the forces to be engaged in the conflict. In addition, the need arises for many other quantitative factors describing the geographic region, weapons capabilities, and many similar data as required by the particular objectives of the study.

While this work is being done, attention must also be focused on developing rules and procedures for the play phase. This includes the rules according to which the players will make their decisions and the procedures by which control will implement the players' orders. Establishment of procedures also includes the development of the assessment models, since these models and the way in

which they are programmed will reflect the decisions made with respect to procedures. In the semiautomatic system, this is perhaps the most time-consuming element of the preparations and also the most critical. Efficient rules and procedures together with realistic models are among the most important aspects of the system.

As the mechanized components of the system are defined, and after the quantitative factors have been obtained, some time must be spent in putting these data in a form consistent with the input requirements of the models. Here, again, effective procedures will ensure less time being wasted during play of the game because of improper or inaccurate data.

Prior to the play of the game, some time must be devoted to player orientation. First, the players must be briefed on the scenarios so that they may become familiar with the environment for the game and learn what is expected of them. They must be given their game objectives. Second, they must be instructed on the rules of play. They must also be given a good understanding of the mechanics of the play so that they may better expedite the system and use it to its full potential. Finally, they must be provided with a record of the status of their forces and all relevant data.

Orientation of the players is the final step prior to the play of the game. Once this has been accomplished, the second phase of the system can be initiated.

Game Play

The game-play phase can be considered as a repetitive cycle of events, the cycle being repeated until the prestated objectives of the play have been realized, i. e. , until one of the player teams has been successful in achieving its predetermined goal. In some cases, however, this may not be possible, and it then becomes the responsibility of the control group to terminate play.

The player teams initiate play by determining what tactics or strategies they wish to employ in achieving their goals. On the basis of their mission and available forces, the players generate orders that are communicated to the control group. The control group then takes the orders issued by both player teams and integrates them while simultaneously judging their relative feasibilities. Control, in rendering these decisions, considers such aspects as whether or not one side's forces can execute its orders without exposing itself to enemy action, or whether or not a move is logistically feasible. Once control has evaluated the orders, it is necessary to specify the interactions that will result. Viewing the execution of both teams' orders with respect to one another, the control group is able to establish what interactions will occur.

As the battle situations become evident, control translates a description of these interactions into appropriate machine language. When all the battles have been so defined, it is then possible to feed this information into the computer. The computer, on the basis of the models programmed during the preplay phase, then assesses the outcomes of the interactions of the opposing forces. It determines what has been gained and lost by the two sides. On completing these calculations, the computer generates outputs that consist of the results of the assessment in terms of casualties, moves, and other similar information. These results are distributed to the two player teams and to the control group. On the basis of the results, the control group prepares a summary of the action for the players to supplement the machine results.

The cycle then begins again, with the players weighing the results against the achievement of their objectives. Based on the current status of their forces and whatever intelligence estimates they may have received, they generate a new set of orders, and the cycle is repeated. This repetition occurs until, as said previously, either the control group halts play, or a player team realizes its goal. When play is stopped, the last phase of the system begins.

Postgame Analysis

Analysis of the game is perhaps the least defined aspect of the system. It can follow a number of different courses, dependent on the original intent of the study; nevertheless, there is a very general pattern that this phase might follow. Many questions must be asked: What were the critical aspects of the game? What caused the turning points of the action? How did the initial situation as defined in the preplay phase affect the outcome of play? It must be determined what essential elements of the game influenced the consequent action and how they affected that aspect of the play relevant to the stated problem. Analysis of these factors can be both quantitative and qualitative. The former lends itself well to being resolved on the computer, whereas the latter most generally is handled by the control group with support from the players. It is important to realize the potentiality of computer analysis of the results. Since the results have all been generated by the machine and complete records kept in machine language, all the data required for a quantitative analysis of results are already in a form suitable for immediate machine analysis.

Interpretation of the quantitative and qualitative analysis leads to the conclusions to be drawn from the game. From such a system, both substantive and methodological conclusions may result. The methodological conclusions are then incorporated into the system, improving it for the next play, whereas the substantive conclusions are either held until numerous repetitions of the game can further substantiate them, or used to infer possible recommendations with regard to the original study directive.

Figure 1 summarizes the material presented in this section. Each aspect of the system that is a separate event is enclosed within a rectangle; also included, in some cases, is a brief indication of the activities performed during the event. The diagram also serves to demonstrate the principle of information

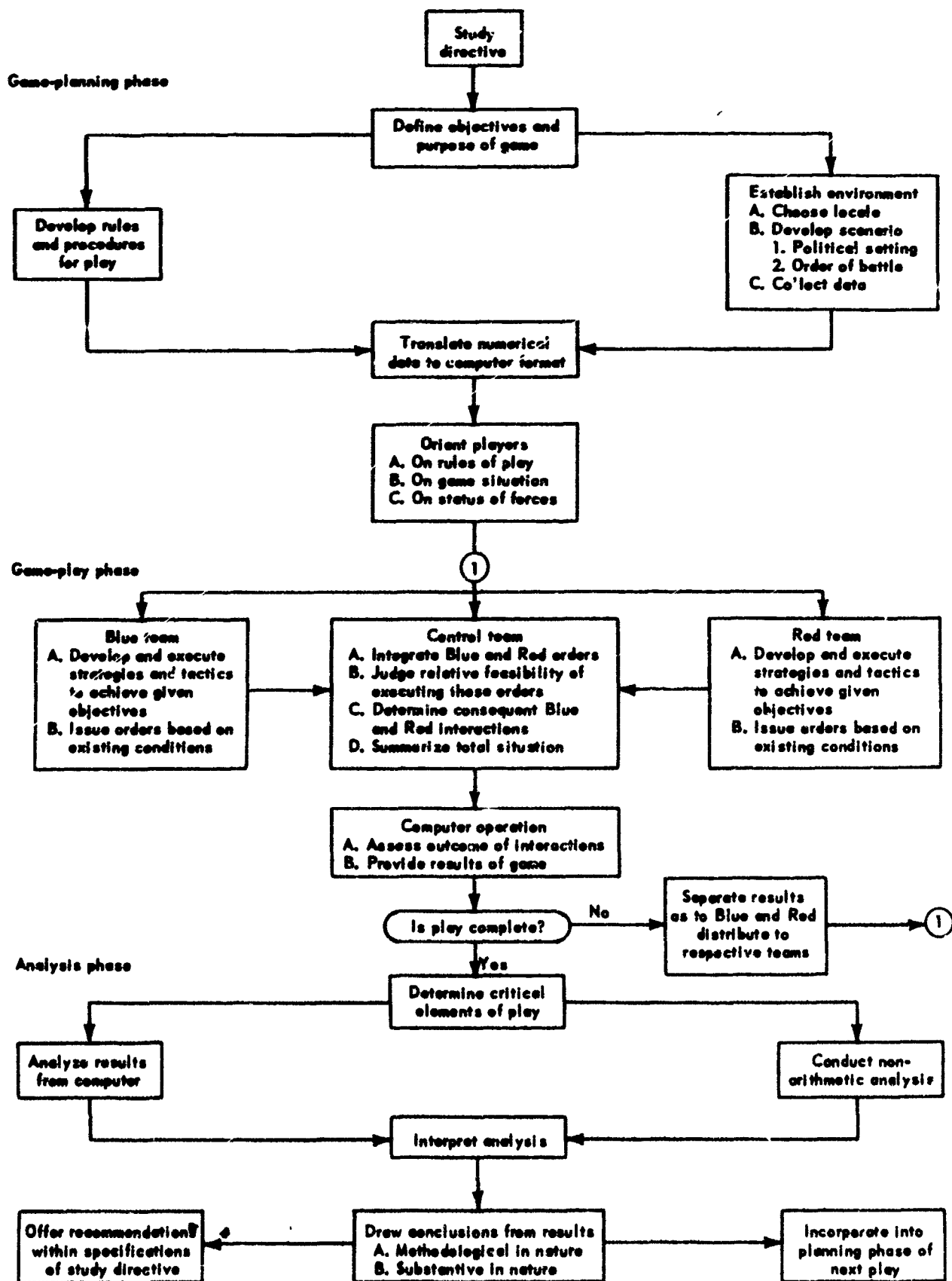


Fig. 1. Sequence of Events in a Semiautomatic Gaming System

flow. In the following section, the reason for the emphasis of this principle will become evident. The computer programs, the subject of the next section, have the primary function of providing for the proper flow of the information necessary for assessment calculations during computer operation.

THE MECHANICS OF THE COMPUTER OPERATIONS

The two basic purposes for using the computer in the semiautomatic system assessment and bookkeeping – have been indicated in the first section. The assessment function is accomplished through the application of various models, defined by the type of function they perform. For example, an air model assesses the interactions occurring during various phases of air operations, such as escort missions, interceptor missions, reconnaissance, interdiction, and the like; there will be as many models as there are well-defined, distinct assessment operations. The need to interconnect these models generates the requirement for some master program that provides the medium in which these models can operate. There is the further stipulation that this master program will be responsible for maintaining accurate and up-to-date records, with the provision for automatic changes to these records.

Thus, it is the intent of this section to enable the reader to understand the requirements of a master program, based on its inputs, its operations, and its outputs. The objective is to describe the characteristics of the master program in a way conducive to other applications, i. e. , so that others may find use for it.

Objectives

The specifications for the design of the master program are to:

- (a) require a minimum control effort in composing input to the computer,

- (b) establish an input format that is meaningful to control (a minimum of symbolism),
- (c) include the means for processing, routing, and storing data sets for use by assessment models,
- (d) allow for the operation of logically distinct models,
- (e) provide a method whereby accurate records may be maintained with the capability for their alteration, and
- (f) enable results to be displayed in an understandable form.

Input

Inputs to the computer falls into three categories:

- (a) that resulting from control definition of the combat interactions,
- (b) the status-of-forces file that includes all units being played in the game and their attributes, and
- (c) those from control that do not result from any defined interaction, but are changes to the status-of-forces file.

The last category includes such changes as increasing the number of men in a unit when reinforcements are introduced by the control group, or specifying a new location when a unit is to have its assigned location changed. (These examples assume that strength and location are attributes of a unit and are recorded in the status-of-forces file.)

The basic principle involved in the input that defines the interactions is that all units participating in a given combat situation will comprise what is termed a "battle group," and the information for each battle group will be recorded on punched cards, one card per unit. All such units must be designated explicitly to be considered by the assessment models. In addition to naming the units, it is assumed that certain factors describing the conditions of the battle

and influencing its outcome would be included. Such factors as posture, terrain, and type of engagement might be included.

Each interaction defined as a separate battle group is processed as a separate engagement within the computer assessment of the outcome. The control group has the responsibility of specifying the different battle groups and parameters involved for each play; the master program maintains each as a separate entity in referencing the assessment models.

One of the fundamental elements in the system is the status-of-forces file. It is prepared, initially, during the pregame planning phase by the control group. All relevant data for each unit to be played in the game are placed on standard forms, and they are then translated and processed onto magnetic tape. This is the only nonmechanized, or nonautomatic aspect of maintaining the status-of-forces file. It then serves as input to the initial interval of play, after which it is automatically revised consequent to the assessments of outcomes, and any new values for the characteristics of units are then incorporated into it. The characteristics of the units contained in the status-of-forces file are an integral part of the determination of the outcomes of the interactions. These characteristics are the factors plugged into the formulas of the models. The emphasis placed on the processing of these data will be seen later in this section.

The last type of input to the computer system is related to the status-of-forces file. As has been explained, this file exists on magnetic tape and is automatically processed and changed by the master program as a result of changes to unit characteristics generated by the models. However, the possibility for nonmachine-generated changes must be acknowledged. For this reason, provision is included within the master program to incorporate changes to unit characteristics issuing directly from the control group. Thus, by control decision, whole units, or part thereof, can be eradicated or revised automatically in accordance with the changes recorded on punched cards.

Description of the Master Program

In the first section, the principle of information flow was emphasized. In terms of the master program, it is of equal importance; however, in the medium of the computer, the information assumes the form of data sets. The input information in raw form is organized by the master program into logically distinct data sets. The master program is then concerned with the ordering and storing of these data sets. When this has been accomplished, the master program references the relevant models that are to operate on the data sets. As changes to data pieces within sets occur, it is the responsibility of the master program to incorporate these changes into the data sets. Finally, when all the changes have been effected, the master program provides the means whereby the revised data sets are edited and dumped as output from the computer. The master program is composed of six basic routines that enable it to accomplish these functions. It is the function of these routines to:

- (a) read battle group cards,
- (b) select and store status-of-forces data,
- (c) reference models and adjust data,
- (d) edit assessment results,
- (e) update the status-of-forces file, and
- (f) edit the status-of-forces file.

Each routine will be discussed in terms of data sets, with regard to the procedures to be followed in executing its operations, the input required, internally stored data necessary for execution, and the results of the operation.

The general flow of operations performed by the master program is presented by the simplified flow diagram in Fig. 2. The more specific details of the operation have been excluded from this chart. In determining what should be included, the authors have attempted to present only those relations which, if

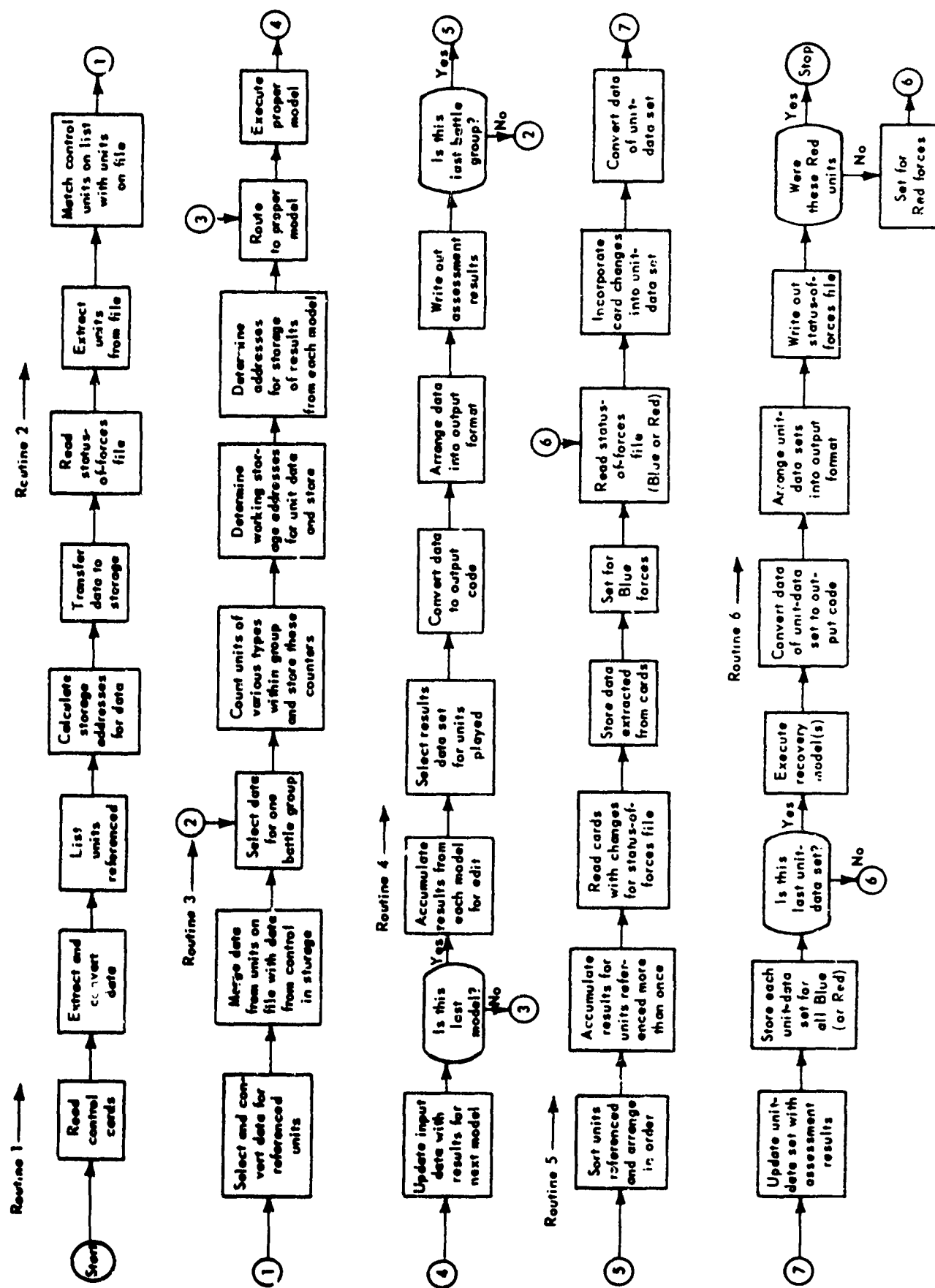


Fig. 2. Master Program: General Flow Diagram

changed, would, in effect, create a different program. It is believed that changes within any one of the individual boxes would not appreciably affect the over-all program; but to change the relation among the operations illustrated would be such a significant alteration that it would be more advantageous to design a new program.

Read Battle-Group Cards Routine

The master program starts by reading in the control information defining the battle groups, or combat interactions. This information has been recorded on cards punched in a specific format designed for the problem. Each card contains the designation of a unit involved in the interaction, in addition to parameters relating that unit to the battle situation. The data are extracted from the cards and are converted from the input code to the internal language of the computer. The process is continued until all the cards for the units being played during the present interval have been read. The names of these units are stored in a list that is to serve as a key for model routing. The control data for these units are then stored to be integrated at a later stage with the information extracted from the status-of-forces file. When all the battle groups have been read into the computer and processed in this fashion, the functions of the first routine have been accomplished, and the computer system is ready for the second routine to begin operation.

Select and Store Routine

The select and store routine also performs an input function. This routine reads in the status-of-forces file (from magnetic tape). Contained on this file, as has been mentioned, is a record of all the units and data describing these units. The routine, in reading the file, checks the name of each unit against the list of names made from the control input, and when a match is found, the data for this unit are extracted from the file. The data are then converted from the tape code to the internal language of the computer in the same fashion as were

the card input data. Once the data have been extracted and converted, they are stored with the card input for the unit. The process is repeated until all the units contained in the control list have been matched with data taken from the status-of-forces file. When all such information has been stored, the storage region will be organized in the form of the following example (in consecutive machine cells):

Name of first unit specified	3d Div
First control input parameter (posture)	Defend
Second control input parameter (terrain)	Flat
First file datum (location)	Berlin
Second file datum (strength)	9000
Third file datum (armament, %)	100

This process is repeated for all specified units. Each unit and its corresponding input data organized in this manner within the computer are referred to as a "unit-data set." The remainder of the explanation of the computer system will be focused on the processing of this basic entity, this process being analogous to the principle of information flow in the nonautomated portion of the system.

Model Selector and Data Adjuster Routine

The central routine of the master program is the model selector and data adjuster program. The other routines of the system merely supplement the functions of this routine. Its purposes are to reference the appropriate model and to provide it with the unit-data sets necessary for its calculations. To accomplish this, the routine first selects the unit-data sets comprising one battle group and transfers these to a working area. Next, it determines what type (and how many) units are represented in the group; by doing this, the routine is then able to determine what models should be called in to assess the outcome of the interaction. At this point, a slight digression is warranted to make explicit the assumptions underlying this approach and what it requires.

The obvious premise is that the type of unit involved in an interaction determines what model should assess its effect on the outcome. Specifically, it implies that a battle group composed only of air units, perhaps squadrons or wings, should be processed by an air model. This is obvious; however, what is not so clear is the procedure to be followed when the battle group is composed of a mixture of types of unit, i. e. , a battle group containing air, artillery, armor, and other dissimilar units. What procedure is to be applied must be decided early in the pregame planning phase and requires what might be considered simply a delegation of responsibility -- which models should assess what portion of the interactions. The approach agreed on by the control group is arbitrary as far as the master program is concerned; regardless of what decision is reached, however, some means of specifying the unit type is necessary. Thus, the two requirements for the master program are that first, a doctrine be defined, and second, that a means be provided whereby it is possible to differentiate between the types of unit.

With this in mind, the reader can now better understand the function of the master program to determine what types of unit are present in the battle group. Before the models can be executed, however, there are still two operations that the master program must perform. It prepares a list of machine addresses, which are the first cells of each type of input unit-data sets, and it also calculates the amount of storage necessary for results of the assessment and assigns storage addresses for this purpose. At this point, the master program is ready to reference each model in turn in accordance with the procedure established in the planning phase.

The master program thus provides each model with the following four items: a) the input data sets, b) the addresses of the locations of these data sets, c) the number of the various types of unit within each battle group, and d) the first addresses of the storage areas where the results are to be placed.

After each model has assessed the interaction and has stored its results in the results region, the master program revises the input unit-data sets with respect to these results so that, as each subsequent model operates, it is provided with an updated data set. In this way, there is established an interconnection between the various models of the system. This also demonstrates the importance attached to the procedure to be followed concerning the order in which the models are to be referenced. Since this is a fixed system, i. e. , the logical order of the models never varies, emphasis should be placed on selecting that order which most nearly represents the usual sequence of events in reality. *

After the models have assessed the outcome of a particular battle situation, the entire assessment operation is repeated for each of the remaining battle groups. At the completion of each cycle, the input data sets for the processed battle groups are discarded, and the data sets of results are stored for the later phases of the operation.

Results Edit Routine

It is the function of the results edit routine to provide the output from the assessments. It first selects a unit-data set of results. Next, it converts the data into the output code, arranges them according to the output format, and writes them on magnetic tape. The results indicate all those items of the status-of-forces file that have been altered by the models and are the actual changes, not the results of these changes. For example, given an infantry division that has suffered heavy losses in combat, the results from this might be the number of casualties suffered to personnel and losses of equipment.

*It is acknowledged that in reality sometimes events occur simultaneously; however, reality must be compromised to be made compatible with the fact that the digital computer operates sequentially.

The results output then consists of the name of the unit and changes to that unit, and these are given for each model and in total for all models. The routine continues in this manner until the results for all units played during the interval have been edited.

Update Routine

It was pointed out at the beginning of the discussion that the status-of-forces file was automatically maintained, and it is the function of the last phase of the system to accomplish this task. The first part of this operation is the update routine. The routine sorts all the data sets of results and arranges them in the same order as they appear in the status-of-forces file. This generates the requirement for a definite order for the units in the file. This could be done in either of two ways: a list of the order of the units in the file could be stored within the routine, or the units could be arranged in some logical pattern in the file. The latter choice is the one incorporated in the computer system; it is assumed that all units are recorded in the file by number and that these numbers are in ascending sequence. Thus, the routine is able to order all the data sets of results in ascending sequence to facilitate the updating process. While ordering these data sets, the routine checks for units that have been referenced more than once. Where a unit does appear more than once in the data sets, the results are accumulated forming just one data set for each unit, further facilitating the updating process.

An auxiliary function of the routine is to provide the capability for making nonmachine-generated changes to the status-of-forces file, i. e., those changes that directly reflect a control decision. To execute this, it is possible to introduce such changes by punched cards. To ensure that computer storage restrictions would impose no limitation on the number of units that could be changed in this manner, the card changes are read for only one unit at a time; the next set

is read in after the first set of changes has been made. These changes therefore, must, be in the same order as the units in the file. Any datum for a unit can be changed, except the identification number. The number of such data changes is unrestricted so that, for example, control could revise the number of personnel assigned to a unit to reflect a decision regarding reinforcements, or it could alter all the data attributes if necessary.

After the results of the model assessments have been ordered and a set of control changes for one unit read in, the routine begins to read in the status-of-forces file from the magnetic tape. As each unit is extracted from the file, a check is made to determine whether any of its data attributes are to be replaced by those data of the control cards. (In the present system, two cards are required per unit.) If there are any, the new data are substituted for the corresponding data comprising the file unit-data set. Next, the unit-data set is converted to the internal language of the computer, and a check is made for the existence of any assessment results for the unit. When such results are present, the file unit-data set is updated with this information, and the revised unit-data set is stored within the machine. This process is continued until all the units for one side (Blue or Red) have been transferred from the file into the computer, at which point the integration of all changes for these units from control and the models should have been completed.

The reason for storing all the data sets of just one side is to provide what is required for the execution of models that do not perform interaction assessment calculations but, rather, that accomplish what might be called "recovery procedures." It is assumed that such models do not, therefore, require access to unit-data sets for both sides; as a consequence, only the data sets representing either Blue or Red units, respectively, are stored at any given time for these models. By this approach, a more effective utilization of storage space

is accomplished. (In the THEATERSPIEL application of the system, a logistics model was included at this point to perform consumption and resupply calculations for all units in the theater of operations.)

Output Generator

After the model, or models, have been executed, the system has only to generate the output. Output is generated after each pass through the update recovery portion of the system, i. e. , after both the Blue and Red units have been processed. This involves selecting, in turn, each of the unit-data sets, converting the data into the output code, arranging the data sets into the output format, and writing this material on magnetic tape; in doing this, the revised status-of-forces file is produced.

From the system, then, two forms of output result: assessment results and a revised status-of-forces file. In addition, all inputs to the system have been placed on punched cards. Thus, all the quantitative material of the play exists in machine language. These three items can be retained for the work on the postgame analysis phase and provide the initial means whereby this analysis can be efficiently executed by the computer, thus accruing an important additional benefit from a computer-supported gaming system.

THE THEATERSPIEL COMPUTER SYSTEM

The computer system described previously has been designed for the THEATERSPIEL Study 35.10, Conflict Analysis Division, for its POMEX war game. The development of the system was accomplished through the joint work of this study group and the computing laboratory staff; as such, many of the decisions concerning critical aspects of this development reflect the efforts and decisions of both groups.

POMEX was played during the latter part of July and the early part of August 1961. In preparing for play (Phase I) and during play (Phase II), a great deal of attention was directed toward providing for the efficient employment of the computer systems; at the same time, much was learned in applying the system. It is the purpose of this section to present some of the methods devised for the application of the computer system by THEATERSPIEL, and later, to give some indication of what one computer-oriented experience has gained for the study group so that other studies with a similar orientation may benefit from this first attempt.

Phase I: THEATERSPIEL Computer-Usage Preparations

It was decided that the computer-oriented objectives for the play of POMEX would be to mechanize four separate models: an air model, a support weapons model, a ground combat model, and finally, a logistics model. The substance of the models, coupled with the over-all objectives of the study, determined the level of aggregation of play, i. e., the amount of detail to be included. As a consequence, the size of units to be played was that of division level. Further, the choice of the particular theater to be played affected the decision as to what types of unit were played. Finally, the data required by the models for each unit determined what characteristics were used to describe the units. It was in this manner that the specifications placed on the design of the status-of-forces file became evident.

The preparation of the status-of-forces file involved several stages of development. First, a unit designation system by which the units could be identified was devised consistent with the various classifications of units to be considered in the game. It was based on the use of either a B (Blue) or an R (Red), followed by four digits which were coded representations for a) the type of unit, b) the nationality, and c) the specific number of the unit (the last two digits).

There can be no more than 100 units of any given type and nationality with the use of this symbolic system; however, the addition of one digit would increase this number to 1000 and would remain compatible with the system in its present form. The unit designation number, as such, was used throughout the game by both the player teams and the control group when referring to specific units. For map purposes, only the last two digits were used on the unit symbols. However, the color and shape of the symbol determined the rest of the designation, and thus, identification of units was accomplished.

After the scheme for unit designations had been developed, it was necessary to specify what characteristics would form the various unit-data sets for each type of unit. Each model required certain pieces of data for each unit. The goal of compactness, however, was kept in mind, i.e., where possible, the pieces of data accumulated served more than one model's needs. Table 1 contains the result of this effort. As can be seen, there are six unit types represented for which, in most cases, the characteristics are the same. For example, each unit-data set, except for supply points which are a special type of "unit," has a characteristic labeled "pres. str.," the abbreviation used for "present strength." This has different meanings depending on the type of unit described: for air units, it describes the number of planes; for SAM units, the number of launchers; and for the rest, the number of men. The versatility of this characteristic demonstrates what was achieved in striving for compactness and applies to many of the other characteristics. An explanation of the meaning of the other abbreviations can be found in Table 2.

In the process of preparing the status-of-forces file, the next step was to design a suitable format. It has been previously indicated that the file exists on magnetic tape; however, the actual working file, which the player teams and the control group use, is the listing made from the magnetic tape on the high-speed

Table 1

Characteristics of Unit-Data Set

Branch	Name of unit	Unit designation	Location	Activity	Road in	Priority	Max in-flight cp	Pres in-flight cp	Tons/100 men	Pres str	Auth str	Air-aid cp	No. sorties	Prev firings	Weapons	Com-bat pot	Max sup strd	Prs sup strd	OH supply				Road out																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
																			1	2	3	4	5	6	7	8	9	10																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
Air	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Table 2
Definitions of Unit-Data Sets Characteristics

CHARACTERISTIC	MEANING
Name of Unit	Actual name of unit
Unit Desig.	Designation number of unit used in game system (explained in text)
Location	Geographic position of unit
Activity	Mission of unit in given interval of play
Road In.	Number indicating supply point from which supplies are to be obtained
Priority	Number indicating relative importance of unit in obtaining supplies
Max. Inpt. Cp.	The upper limit (or original value) of a unit's capacity to receive supplies
Pres. Inpt. Cp.	Capacity of unit to receive supplies during given interval
Tons/100 Men	Total authorized weight of unit per each 100 men
Pres. Str.	Number of men in unit available for combat at end of given interval
Prior Str.	Number of men in unit at beginning of given interval
Auth. Str.	Original number of men assigned to unit
OH Supply	1. Total weight in tons of class 1 supplies unit has available 2 and 4. Total weight in tons of class 2 and 4 supplies unit has available 3. Total weight in tons of class 3 supplies unit has available 5. Total weight in tons of class 5 supplies unit has available
Airfld. Cap.	Number representing the capacity of an airfield to expedite air operations
N . Sorties	Number of planes flown during last interval
Prev. Firings	Number of missiles launched during last interval
Weapons	Index of combat value for support weapons units
Combat Pot.	Index of combat value for ground combat units
Max. Sup. Strd.	The upper limit on amount of supplies a supply point can store
Prs. Sup. Strd.	The amount of supplies a supply point has stored during given interval
Road Out, 1-10	A number indicating which units can obtain supplies from given supply point

printer. The format was designed with this fact in mind, and also considered the size of the sheets of printer paper, the amount of material on the status-of-forces file, and the clarity of presentation. The result was a printer page containing, at most, eighteen unit-data sets arranged in two tables of nine units each, with the data pieces of each unit-data set placed vertically with respect to one another. An example of this format is provided in Fig. 3.

For the other two inputs to the computer, different formats were used. Figure 4 shows a sample card format for the battle groups. The control group filled out similar sheets to be keypunched on cards and read into the machine. The first six columns of the card contained the unit designation number; the next six were used to specify the percentage of the unit being employed in the case of air units or, in the case of ground units, to indicate the type of terrain in which the engagement was to take place; the third set of six columns was used to specify the type of engagement being fought, and the fourth set of six columns was used to indicate either the target for an air unit or an alternative location for a ground unit. The remaining 56 columns were reserved for comments, and although these were not a necessary part of the computer input, they were used by the control group to supplement the records. The repetition of the number six is significant with regard to the input and output of the system, and there is a reason for it. The input-output is written in the specific code, or language, required by the high-speed printer, which permits the intermixing of alphabetic and numeric characters; as such, it is only necessary for working with the magnetic tapes (the status-of-forces file and the assessment results), but for consistency, it was decided to employ the same code when using cards. In this way, the entire input-output medium is written in the one language; all pieces of data included can consist of no more than six characters due to translation of these data pieces into internal machine language and the given word size within the computer.

Fig. 4. Sample Card Format for Battle Group Input

This same idea applies to the status-of-forces file changes made by the control group. These changes are punched on cards, and each six columns on the card corresponds to one data piece of the unit-data set. Since each unit is described by some twenty data pieces, two cards are used per unit (columns 79 and 80 are omitted). For example, columns 13 to 18 would contain whatever new data piece that should replace the third data piece of the unit-data set (counting the unit-designation number as the first). Suppose that control desires to change the "road in" to the Russian armored division designated R3601 (see Fig. 3). A card would be prepared with the designation punched in columns 1 to 5 and the new road-in number (512) in columns 16, 17, and 18. The revised status-of-forces file for D + 9 would then reflect the change.

In addition to designing the input formats, one output format had to be prepared for the results of the assessments. The sample format in Fig. 5, shows the results for three units. For each unit the format shows its name and designation number, the date, its new location (if it has been moved, as each of the three have), and the various losses of the unit. The casualties are given in four columns - the first three show the losses calculated by the assessment models (air, support, and combat) and the fourth the total casualties. Each page contains units of only one side, either Blue or Red; this is done so that the results can be separated and distributed to the respective player teams. Every unit specified within the battle groups in any given interval of play will be included on the results sheets in this fashion.

During Phase I, the four models were programmed and integrated with the master program. A great deal of care was taken in the coordination of this task to ensure internal consistency within the resulting computer system. Each model was written as a separate program which, when finally incorporated in the complete system, could be entered from the master program and exited as an internal, logically distinct and independent program, requiring only the

CASUALTY ASSESSMENTS POMEX I				
	8 TK DIVISN NEW LOCATION		R3608 NB80	D + 8
CAUSE OF CAS	AIR	ARTY OR SSM	GROUND	TOTAL
INPUT CAP	2365	0	0	2365
PERSONNEL	163	131	539	833
COMBAT POT	7	6	23	36
	4 MTRZDIVISN NEW LOCATION		R3623 NA76	D + 8
CAUSE OF CAS	AIR	ARTY OR SSM	GROUND	TOTAL
INPUT CAP	1503	0	0	1503
PERSONNEL	166	152	628	946
COMBAT POT	3	3	13	19
	5 MTRZDIVISN NEW LOCATION		R3624 NA78	D + 8
CAUSE OF CAS	AIR	ARTY OR SSM	GROUND	TOTAL
PERSONNEL	N	159	656	815
COMBAT POT	O	3	15	18
	N			
	E			

Fig. 5. Sample Assessment Results Format

input data external to itself, i. e., the unit-data sets, the initial addresses thereof, and initial addresses for the results storage. As this programming effort and the data preparation were completed, attention was directed to determining what procedures would be followed in the execution of play.

Phase II: An Illustration of THEATERSPIEL Game Play

The purpose of this section is to illustrate how all the elements of the game fit together during its execution. To accomplish this, an artificial situation will be developed in which all aspects of the game cycle are included. In this situation, the prescribed mission for Blue is to block the enemy advance and to defend at all points along the main line of resistance. The Red force has been ordered to make a rapid penetration and seize all river crossings in the southern sector.

In support of his mission, the Red commander provides control with a message stating that two regiments of the 7th Armored Division together with two regiments of the 8th Armored Division should proceed with all due haste to secure a foothold at the crossing now held by the Blue 101st Airborne Division. Blue, having received intelligence estimates on this situation, issues instructions to the commander of the 101st to hold until an armored division can reinforce his position.

The messages from both teams are delivered to the vice-controller in charge of combat operations. A part of the process for effecting the revised deployments includes consultation with the assistant controller for logistics, who determines that the river crossings are not sufficient to support the main thrust of Red's advance. In the light of this evaluation, the vice-controller permits only the forward elements of the Red 7th to make the initial assault. In so doing, they come into direct contact with the Blue 101st. The vice-controller, noting the proximity of a Blue artillery battalion to the site of the

engagement, brings to bear his supporting fires in defense of the Blue position. In this way, the resulting combat interaction is defined. In final evaluation of the action, he judges that Blue has been in position for a sufficient period to allow the preparation of suitable defenses. The combat situation is then summarized as one in which Red is attacking Blue in a prepared position in an environment in which the terrain is unsuitable to extensive use of armored vehicles – one of the cases the computer is programmed to evaluate.

Following the instructions received from the Red and Blue players, now modified by the vice-controller for combat, the air operations controller commits the Red 1st Fighter-Bomber Squadron in support of the Red offensive. Since no Blue air is available in the area for defensive purposes, Red air is able to interdict the position of the Blue 101st unmolested.

While the controllers responsible for the combat phase of the operations are performing their evaluations, the assistant controller for logistics is preparing the information necessary to the operation of the logistics model. First, he must determine how much supply is to arrive at the theater terminals, i. e. , what the intratheater LOC is capable of sustaining at the present time. The amount is recorded for use as input to the mechanized model for distribution throughout the theater. In the process of performing his routine examination of the condition of the intratheater LOC, he discovers that the same 101st Airborne, if it is to sustain combat effectiveness during any prolonged period of combat, must receive some extra supply by airlift. Discussion with the air operations controller results in suitable arrangements to effect this on the following day, when a Blue transport wing in the area will have one squadron of aircraft available for the operation. Since Blue has requested that one armored division be advanced in an effort to reinforce the 101st Airborne, the logistics controller begins the process of administratively moving the unit forward. This,

too, must be prepared as input to the programmed model in order to place the proper constraints on the capacity of the intratheater LOC.

Once the decision-making phase of the control operation is completed, it is necessary to begin transcribing the symbolic representation of the consequences of these decisions onto forms from which punched cards can be prepared. The clerk for ground control interprets the information provided by the vice-controller. This includes the units involved, their adversaries, the type of engagement, and the required battlefield parameters. In a similar fashion, the information provided by the air operations controller is coded onto input forms, as is the input data for logistics.

The various input forms are collected and given to the controller for machine operations. The air mission forms are merged with those of the ground combat. The controller checks for consistency to ensure that proper coordination is reflected in the coded information from the respective controllers. This check is made to prevent the incorporation of any human errors in the input, e. g. , to confirm that all rules for input preparation have been observed. Any discrepancies detected are eliminated, and the final result is confirmed with the individuals concerned.

After the controller for machine operations approves the forms, they are routed to the key punching staff of the computing laboratory. The information is punched on cards which are then verified and returned to the controller. The punched cards are arranged in the order of the program input specifications. The cards describing the combat interactions are placed first, followed by the input for the logistics model. The ordered deck of cards is listed for a final visual verification and for the record.

The computer phase of the cycle is the simplest phase of the operation. The information punched on the cards is read by the assessment program. The

battle is fought and the assessment is performed. The casualty assessment results are printed on the high-speed printer, while the logistics model continues to perform its calculations of consumption and resupply. A revised status-of-forces file is the final result.

The output is returned to the control room to be sorted, separating the Blue and Red results for distribution to the respective player teams. The various controllers supplement the output with written reports summarizing the action of the day. The vice-controller reports to Blue that his 101st Division has held but has taken heavy losses, as shown by the casualty assessment output. Red is informed that he has succeeded in inflicting heavy losses, but that he has not succeeded in securing the river crossing.

Red finds that he must decide whether to continue the present assault or to delay until the two regiments of the 8th Division can be added to his attack. Meanwhile, Blue must decide whether to hold and continue to take heavy losses, or to begin an orderly withdrawal. The cycle begins anew.

The play continues in this manner, with the players using the revised status-of-forces file each time together with the other information given them by control group to generate their new set of orders, until the controller judges that the combat has achieved the degree of resolution that had been initially desired, at which time the game is ended.

This description represents one of the many possible approaches to the game-play phase. The approach will vary with the situation. No rigid pattern should be established; flexibility is necessary if the many unusual situations which frequently arise are to be met and resolved. For this reason, the above exposition should be accepted only as an example and not as the general case.

This short summary of the THEATERSPIEL computer play of POMEX has been included to indicate how the play of a semiautomatic system can be implemented; in addition, it has been written to provide the background for a discussion of some of the problems realized in such an attempt.

SOME CONSIDERATIONS ON THE SUBJECT OF COMPUTER USAGE

Effects on Game Organization

The effects of computer usage on organization are felt in several ways. First, it becomes necessary to define precisely the rules by which play is to be conducted. The rules must be well defined so that the programming can reflect them. Since the nature of the programming is the expression in symbolic language of a logical progression of operations, the rules must be stated in a form conducive to the accomplishment of this task. Second, the procedures to be followed in implementing the computer system during play must be equally explicit. There must be a specific delegation of responsibility within the control group, and each member of this group must understand at least the fundamental principles of the computer and of the specific programming involved. This is of importance if the maximum potential of the computer is to be realized. If the operation of the system is to proceed efficiently, the members of the control group should have a working knowledge of the programs used. Superficially, this may not appear to be very necessary; however, during the progress of play many unexpected problems will arise, and the individuals concerned must be prepared to cope with these rapidly. Furthermore, an attempt should be made to provide simple and concise forms for each step of the control operations in order to avoid delays arising from errors and from misunderstandings.

There are two further implications which, broadly speaking, can be considered a part of the organizational aspects of a semiautomatic system. As the computer system increases in complexity, it becomes more difficult to revise.

However, if its development proceeds in a logical and orderly fashion, this will tend to alleviate the severity of this problem. Care must be taken to avoid the creation of a black box that becomes unmanageable. Finally, it must be realized that the use of the computer introduces new responsibilities into the control room. One reason for using the computer is to absolve the control group of the many repetitive and tedious functions usually associated with control procedures in a hand-played game. Nevertheless, in diminishing the magnitude of these efforts, it is possible to create new and more tedious difficulties related to the use of the computer, since it creates the requirement for high standards of accuracy.

The Question of Accuracy

The matter of accuracy in working with a computer is two-sided. First, information prepared for the computer must attain high standards of accuracy. Errors made in preparing input for the computer can cause the system to fail in its operation, resulting in unnecessary delay; or the errors can go unnoticed, with the consequence that they are perpetuated into successive stages before they are detected. Here, again, proper organizational procedures can alleviate the difficulty; it must be realized, however, that the final responsibility in this area lies with the personnel of the control group, further emphasizing the need for their proper understanding and knowledge of the system.

Second, the computer provides the means for greater reliability in the accuracy of the results of the game. (This is not to be confused with validity.) Once the programs have been properly checked out, there need be no concern for errors in the computations. Furthermore, the speed and capacity of the computer allow for more comprehensive calculations. Not only does the computer enable the system to include more elaborate methods of calculation that would be unfeasible when performed by hand, but it also allows many more factors to be considered, and in greater detail. Of course, this opportunity

should not be unnecessarily exploited; it is possible to design a system that provides too much detail. If a player team is given an overabundant amount of information, including much irrelevant material, some of it will tend to be ignored and will be of no use. Another aspect to be treated in a cautious manner is the temptation to compromise the game rules, or the corresponding calculations within the models, to adjust to the requirements of the computer. Frequently, certain calculations may present a problem in their translation to a programmed sequence. In resolving the difficulty, the programmer must avoid an arbitrary compromise for the sake of programming clarity. In addition to this, certain operations may arise for which the programming approach is not immediately evident. An approximation of the operation must be performed with an appreciation for the error introduced. Further, it must be realized that this error may cancel out or may become intensified during the remainder of the calculations. From this discussion, the reader should be aware that the problem of accuracy can work for or against the system, although it will generally be a positive factor if the proper attitude is adopted with regard to organization and procedures when designing the system.

In summary, there is a certain danger to be avoided in the use of a computer in the gaming environment. It must be remembered that the function of the computer in this environment is to support the control operations. If the computer receives too great an emphasis, its very advantage can be turned into a detriment, with too little thought being afforded to substance and too much to method. Yet, if concentrated attention is directed toward the design of the system in the pregame planning phase, the degree to which the game play becomes subordinated to the computer operation is reduced to a level at which an efficient relation between man and computer is achieved.

Applications

The system described in this paper could be used to provide a satisfactory approach to any gaming environment similar to the one outlined previously. This is one in which there is a basic concern for the quantitative assessment of the interactions among some fundamental entities, and one in which there is a need for the consideration of a great number of these entities and quantitative factors describing them. Further, there should be a requirement for logically distinct operations to be performed in the calculation of these interactions, which must be repeated a sufficient number of times to warrant their mechanization. Finally, there should be a desire to maintain the separation of the human decision functions and the quantitative analysis resulting from these decisions, and yet there should be the desire to maintain the interrelations involved.

If these conditions are met, then the semiautomatic system discussed could be utilized in any one of three ways. At the first level of utilization, the method of approach might be applied to other studies, thus substantially reducing the necessary planning effort involved, i. e. , it could be applied as a logical system. At the second level of application, the interpretation of this logical system, the master program, could be adapted, with a few slight revisions, to other systems into which the pertinent models could be incorporated. In this case, it would only be necessary to construct these models in a fashion consistent with the requirements of the master program maintaining the basic operations outlined previously. Finally at the third level of utilization, the whole system could be used in toto, including the models programmed by THEATER-SPIEL.

Limitations

Two major limitations are to be considered when discussing the feasibility of this approach to war gaming. Some mention was made previously of the

Applications

The system described in this paper could be used to provide a satisfactory approach to any gaming environment similar to the one outlined previously. This is one in which there is a basic concern for the quantitative assessment of the interactions among some fundamental entities, and one in which there is a need for the consideration of a great number of these entities and quantitative factors describing them. Further, there should be a requirement for logically distinct operations to be performed in the calculation of these interactions, which must be repeated a sufficient number of times to warrant their mechanization. Finally, there should be a desire to maintain the separation of the human decision functions and the quantitative analysis resulting from these decisions, and yet there should be the desire to maintain the interrelations involved.

If these conditions are met, then the semiautomatic system discussed could be utilized in any one of three ways. At the first level of utilization, the method of approach might be applied to other studies, thus substantially reducing the necessary planning effort involved, i. e. , it could be applied as a logical system. At the second level of application, the interpretation of this logical system, the master program, could be adapted, with a few slight revisions, to other systems into which the pertinent models could be incorporated. In this case, it would only be necessary to construct these models in a fashion consistent with the requirements of the master program maintaining the basic operations outlined previously. Finally at the third level of utilization, the whole system could be used in toto, including the models programmed by THEATER-SPIEL.

Limitations

Two major limitations are to be considered when discussing the feasibility of this approach to war gaming. Some mention was made previously of the

storage problem within the computer and of the difficulty involved in changing the routines as the programming becomes progressively more complex. The latter is by far the more important. For example, if it were desired to revise and improve one of the models, the chances necessary would more than likely affect the other models. This snowballing effect would vastly increase the time required to make the alteration. The greater the departure of the new requirements from the original ones, the more difficult the task of adjusting the system to suit these new requirements will be.

The problem of internal storage can be solved, although at the expense of operating efficiency. Extensive use of magnetic tapes can provide an almost unlimited storage capacity for the system. In doing so, however, it must be realized that the speed with which the operations can be accomplished will consequently be compromised, since tape storage, as opposed to internal storage, has a much slower access time for computational purposes. As currently designed, the THEATERSPIEL semiautomatic system's use of tape storage is minimal, with the result that one complete computer run of the system is accomplished in about 15 minutes. It can be anticipated that a significant increase in the use of tape would double or triple this time. Moreover, the reliance on greater usage of tape storage would entail revising the logic of the presently programmed system which in itself would require some time (on the order of months) to accomplish.

Evaluation

In general, there are two major criticisms made of war gaming as a means of problem solution: one is with respect to timeliness and the other with regard to cost. It is the purpose of this section to demonstrate that, in terms of these two criticisms, the semiautomatic system is a significant advancement in the state-of-the-art. There are frequent references in the literature

about the expense that gaming entails, and, furthermore, that it is such an extensive and time-consuming operation that by the time the game is finished and the study completed, there is no longer a requirement for the results. THEATERSPIEL's first play with the system in POMEX would tend to support the view that, by the use of the semiautomatic system, this need not be the case.

After the initial period of familiarization and orientation to the use of the system, the last ten intervals of play of POMEX were completed over a span of two weeks. A hand-game with a degree of complexity and detail similar to that permitted by the use of the computer would require much longer to play. The preparation of the computer gaming system took about six months, including design planning, data collection, programming, and debugging. However, much of the same work required in preparation for play with the computer-assisted system would also be required in preparing for hand-play. The same data would have to be obtained, the same models prepared for use (though in a different form), and many of the same procedures would have to be devised.

Another significant difference to be noted between these two approaches has a bearing on the criticism of timeliness. It has been pointed out earlier in this paper that the postgame analysis can be made for more efficient and profitable if the computer is put to good use. This is especially true when the computer-assisted system has been used, since all the data to be analyzed already exist in machine form on punched cards and magnetic tape. Thus, the problem of organizing the game data for analysis purposes can greatly be diminished by the use of the computer-assisted system.

The advantages gained during the game play phase and postgame analysis phase would seem to more than compensate for the lengthening of the pregame planning phase. Where only one game play with the system is desired, the merits of

the computer-assisted game are not particularly obvious, but repeated use of the computer-assisted system will increase its economic advantages.

CONCLUSIONS

The initial play of the computer-assisted system indicates:

- (a) the original investment required for the development of the computer-assisted gaming system may be no more than required for the development of a manual gaming system having the same degree of complexity, and
- (b) the time and effort devoted to the development of the computer-assisted gaming system will begin to show a return when repeated plays of the system become feasible.

SECTION III

A DATA TRANSMISSION STUDY

This part of the paper reports the results of an experiment conducted to test the feasibility of supporting Army war gaming by a general-purpose digital computer at a remote location. The experiment consisted of the assessment of the air operations and air defense portions of the 1961 - 1962 USAWC war games in Carlisle, Pa. , by the RAC computer in Gaithersburg, Md. Communication between the two locations was accomplished by an automatically encrypted, off-line, card-to-card data transmission system with one fixed and one mobile terminal. The system employed commercial card transceivers, leased telephone lines, and special government-furnished cryptographic equipment.

The experiment was undertaken to investigate the feasibility of applying such a system both for the RAC's use and for possible Army-wide use. A community of interest in machine war gaming of Army problems exists among the Continental Army Command, the Strategic and Tactical Analysis Group, RAC, and the USAWC, and, to a lesser extent, some of the other Army service schools. The existence of a means for automatically exchanging computer inputs and outputs for war gaming might help to further this common interest.

Preliminary work on the sample data transmission system was undertaken in the summer of 1961. It involved, among, other things, coordination with the USAWC concerning participation in the USAWC 1961 - 1962 war games and consultations with the U. S. Army Strategic Communications Command concerning the feasibility of the system and the time required for its implementation. The study was approved and formally initiated in August 1961. The components of the system were assembled at the RAC computing laboratory in Gaithersburg in November 1961. The remote terminal was moved to Carlisle Barracks, Pa. , and tested in December. The system was used to support the 1961 - 1962

USAWC war games in January 1962. The remote terminal was then moved to Gaithersburg, where the two terminals were connected back to back. In this configuration the system was demonstrated to interested Army personnel during the week of January 29. The system was discontinued on Feb. 4, 1962 and its various elements were disposed of subsequently.

RELATED ACTIVITIES

War gaming data have been transmitted between remote locations several times before. The Operations Research Office and the Rand Corporation jointly played a sample manual war game over leased lines between Washington, D. C. , and Santa Monica, Calif. , in 1955. In this earlier exercise, a government-furnished paper tape-to-paper tape system with off-line encrypting was employed. The system was discontinued shortly after the test, when the cooperative war gaming effort was dropped.

In the springs of 1956 and 1957, the George Washington University Logistics Research Project supported the U. S. Naval War College (USNWC) war games at Newport, R. I. , by processing logistic data on its logistics computer in Washington, D. C. The two installations were connected by a paper tape-to-paper tape system operating over leased lines. No classified data were involved. The system was discontinued in 1958, when the concept of the USNWC war games was changed so that quantitative logistic assessments were no longer required.

SPECIFICATIONS

For the USAWC war games, it is necessary to make the result of previous assessments available to the game participants for planning the next action. It was important, therefore, that the total turnaround time for an assessment, including data preparation, two-way transmission, computation, and print-out,

be short -- of the order of several minutes, if possible. This requirement dominated the design of the system, taking precedence over considerations of high capacity. General specifications were:

- (a) So far as the computer was concerned, the system was to be off-line, capable of operating independently of the computer, and storing both input and output data at the computer terminal until called for. It was believed that the lower cost together with the increased flexibility in scheduling afforded by an off-line computer would more than offset the faster data turnaround time that could be achieved with an on-line computer.
- (b) The system was to be a card-to-card system. There were several reasons for this:
 - (1) The 80-column punch card is a universal input-output medium for general-purpose digital computers. The cards may be read by any computer -- for example, STAG's 7090, as well as by RAC's Univac Scientific 1103A computer.
 - (2) Punch-card terminal equipment and mod-demod units to adapt this terminal equipment to telephone and telegraph lines are available commercially and are in wide-spread use.
 - (3) A system employing punched cards and commercial voice circuits appeared to have sufficient capacity for the immediate applications visualized and was much more economical than available higher performance systems.

- (4) Punch cards provide the most convenient and rapid generally available means of preparing and correcting manually generated input data.
 - (5) Punch-card output may be listed by a line printer, whereas competitive paper tape systems employ tape-controlled typewriters. The line printer is better balanced with the rest of the system from the viewpoint of speed.
- (c) The system was to be secured to protect all classified data before transmission. One terminal of the system was to be mobile.

This last condition was imposed to enable the system to be used either to connect one of the other RAC buildings to its computing laboratory or to connect the USAWC to the RAC computing laboratory.

DESCRIPTION

The components of the data transmission system and its associated data preparation and processing equipment were assembled into a configuration of four stations (Fig. 6). The four stations are described below.

Mobile Data Preparation and Print-Out Station

This station consisted of several items of punch-card equipment housed in the RAC trailer van. These items were: a) an IBM 026 printing key punch used to transcribe data for processing by the system; b) an IBM 056 verifier used to verify the accuracy of the keypunching, and c) an IBM 407 tabulator used to print out results. One end of the van was furnished as a waiting room for persons waiting for data to be processed. The van also contained a partitioned routing box for output data being held for pickup.

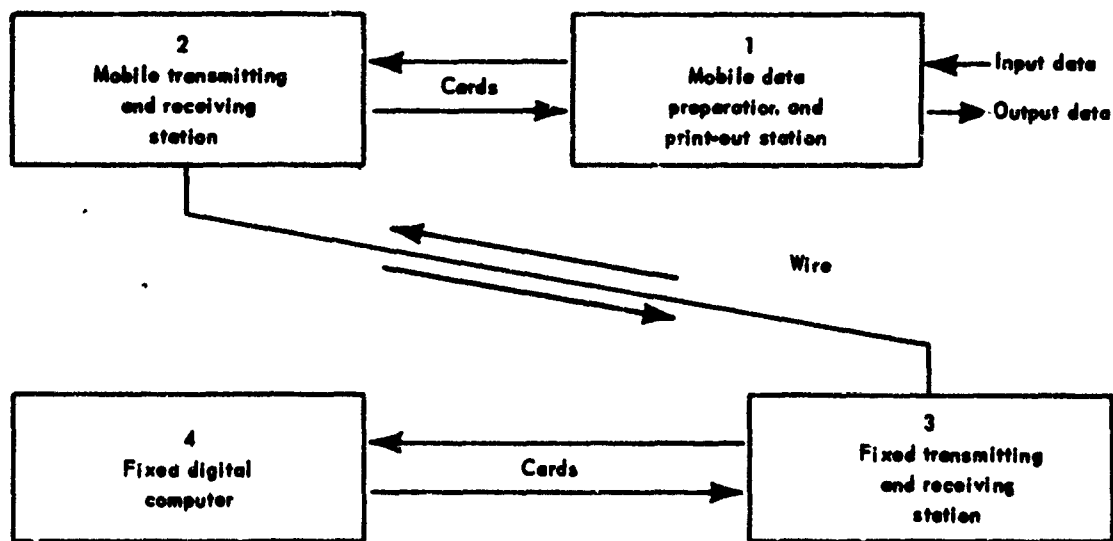


Fig. 6. Sample Data Transmission System Configuration

Mobile Transmitting and Receiving Station

This station consisted of card transmitting and receiving equipment (an IBM 066 data transceiver and an IBM 068 telephone signal unit), on-line cryptographic equipment, and associated circuitry located in a metal hut mounted on an Army 2 1/2-ton truck. The transmitting and receiving equipment operated in the simplex mode, with the same equipment performing both the transmitting and receiving functions, but not at the same time. A voice instrument permitting clear telephone conversation over the leased line was connected to the system. When operating, the truck carrying the hut was backed up to the trailer van so that the hut was accessible only from the van. This facilitated the manual transport of punch cards between the two stations.

Fixed Transmitting and Receiving Station

This station was installed in the RAC Bradley building in Gaithersburg, adjacent to the RAC computing laboratory. The equipment for this station was identical to that of the mobile transmission and receiving station.

Fixed Digital Computer

This was the RAC Univac Scientific 1103A computer. For this experiment, it exercised its capability for reading input data from punch cards and producing output data on punch cards - a characteristic shared by virtually all available general-purpose digital computers.

The mobile transmitting and receiving station and the fixed transmitting and receiving station were connected over leased telephone lines.

The flow of data through the system is shown by the arrows in Fig. 6. Input data were received at the mobile data preparation and print-out station. The data were punched into cards and verified. The verified cards were manually transported to the mobile transmitting and receiving station and read by

the data transceiver. The information from them was automatically encrypted, transmitted to the fixed transmitting and receiving station over the leased line, automatically decrypted and punched in other cards, character by character, to yield a duplicate deck of input cards at the fixed transmitting and receiving station. The duplicate cards were manually transported to the computer and processed. During the manual transportation and computer processing of the duplicate input cards, the transmitting and receiving stations changed mode - from sending to receiving mode for the mobile station, and from receiving to sending mode for the fixed station. The computer output was obtained on punch cards, which were transported manually to the fixed transmitting and receiving station and read by the data transceiver. The information from them was automatically encrypted, transmitted to the mobile transmitting and receiving station over the leased line, automatically encrypted and punched in other cards, character by character, to yield a duplicate deck of output cards at the mobile transmitting and receiving station. These duplicate cards were manually transported to the mobile data preparation and print-out station, where they were printed on the tabulator.

The transmission of each card was checked automatically for illegal character codes and various mechanical failures of both the transmitter and receiver. Each received card satisfying all checks was automatically punched with a 12 in column 81. Transmission was then automatically initiated for the next card in the transmitter hopper. When a card was received incorrectly, the 81st column was not punched, indicator lights were actuated at both the transmitting and receiving stations, and transmission of cards was stopped. The operators then ascertained and corrected the cause of error or retransmitted the card.

APPLICATION

The USAWC war games for 1961 - 1962 were to be conducted by student teams as feasibility tests of plans using war-gaming techniques. Seven plays were to be conducted simultaneously in four different parts of the world. All plays would use the same procedures regardless of location. The assessments were to be performed by the seven student control teams except for specific well-defined portions delegated to the RAC computer via the sample data-transmission system.

The bulk of the air operations and air defense portions of the assessment were prepared for execution by the computer. These assessments constituted a reasonable test for the data transmission system since: a) they served the important purpose of executing the calculations with which the war gamers had previously had the most difficulty; b) the quick turnaround time achieved permitted the execution of successive interdependent air strikes; c) considerable repetition was involved - seven simultaneous plays with repeated assessments of the same type within each play, and d) the calculations to be performed were well defined in advance.

Special forms were designed for communication between the control teams and the computer: a) a ground-air-assessment input form: b) a naval-air-assessment input form, and c) an output form common to both assessments. Card layouts compatible with the forms were devised. One card each was found to be sufficient for the data given in the ground-air-assessment input form and the common output form, but two cards were required for the data from the naval-air-assessment input form. Thus, the processing of a ground-air assessment required one card in and one card out, and the processing of a naval-air assessment required two cards in and one card out.

The computer program was arranged in such a way that an arbitrary number of assessments could be processed as a batch. The program consisted of about 2500 machine instructions exclusive of subroutines, and its speed was limited by the machine's card-handling rate -- approximately 120 cards/minute for either reading or punching.

Three of RAC's programmer analysts completed the program in about two months' elapsed time, including a) conferences with USAWC personnel on firming up details of the program, b) design of input and output forms and cards, c) design of a special tabulator plugboard, and d) debugging. Less than six man-months was involved in all. The program was checked out and operational by the middle of November except for the few ever-present minor bugs that were not detected until over-all systems testing began.

The completed program is now available in 1103A machine language. To execute it on a different type of computer, it will be necessary either to re-program it or to run it on an 1103A simulator. The flow charts, forms, and card layouts are relatively independent of computer type, but may require some modification depending on the computer selected.

IMPLEMENTATION

A number of problems were encountered, both technical and administrative. The technical problems were concerned with designing the system and making it work. The administrative problems were concerned mainly with the acquisition of equipment and satisfying security requirements. The whole administrative process was somewhat cumbersome because of the large number of organizations and agencies involved. The U. S. Army Signal Corps has the responsibility for supplying leased lines and terminal equipment throughout the Army. In this experiment, the Signal Corps appointed an action officer who coordinated not only the work of the various signal agencies, but also the contributions of the National Security Agency and the U. S. Army Security Agency and the acquisition of lines from the telephone company and terminal equipment from IBM.

The technical problems were made difficult because a) the particular equipment configuration selected had not been employed before in the Army, and b) time was limited.

Security problems were of several types:

- (a) Physical security requirements. Some items of special materiel on each end of the line had either to be kept under continual surveillance or secured in a double-locked vault.
- (b) Custodial responsibility. Two custodians, one at each end of the line, had to agree in writing to hold themselves personally responsible to the U. S. Army Signal Communications Agency for the custody of special materiel.

At the beginning of the experiment, a detailed timetable was constructed. It called for a) completion and testing of the computer program by the middle of November; b) delivery, installation, testing, and inspection of all equipment and training of operators by 1 December; c) an operational test of the system between RAC buildings by 8 December; d) movement of the mobile terminal to Carlisle and an engineering test of the system in that configuration by 18 December; e) an operational dry run of the USAWC problem on 3 January; f) operation of the system in support of the USAWC games during the week of 15 January, and g) return of the remote terminal to Gaithersburg for back-to-back demonstrations of the system by 22 January.

The schedule was followed with no major modifications.

PERFORMANCE

Theoretical continuous data rates for the system are shown in Fig. 7 - keypunching and verification at 1 card/minute, transceiving at 11 cards/minute,

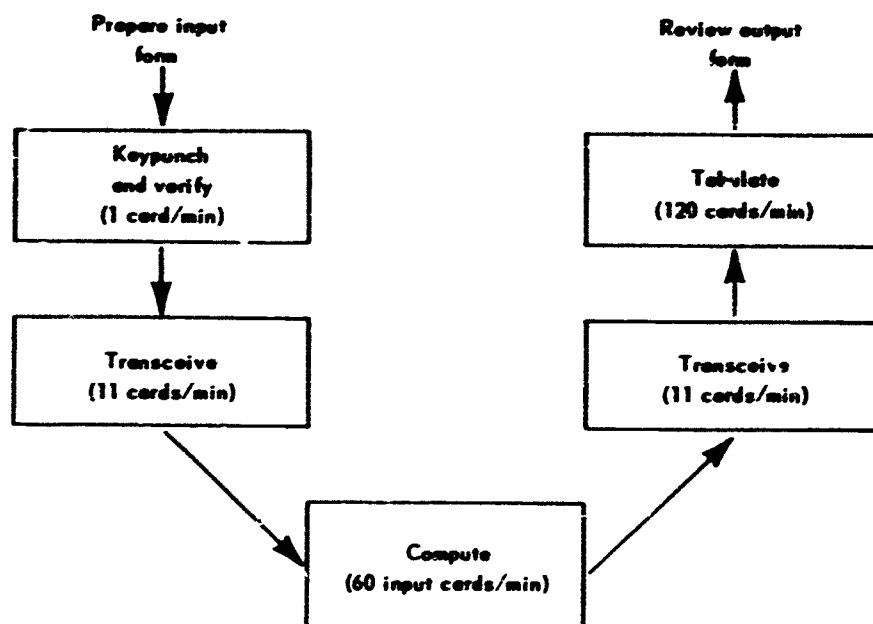


Fig. 7. Theoretical Data Rates

computing at 60 cards/minute, and tabulating at 120 cards/minute. The computing rate is based on a card-limited computation producing one card of output data for each card of input data and, hence, effectively operating at the computer card-handling rate divided by two. If already prepared input forms were put through the system continuously, the over-all rate would be limited by the key-punching and verification rate of 1 card/minute. If prepunched and preverified cards were put through the system continuously, the over-all rate would be limited by the transceiver rate of 11 cards/minute.

In practice, these theoretical rates cannot be achieved. They can be approached for large batches of data. Manual card transport operations (represented by the arrows in Fig. 7, manual start-stop operations for the various items of equipment, and manual changing of the modes of the transceivers are required for each batch of data. Thus, the actual practical data rates depend on the size of the batch of data and the efficiency with which the manual operations are carried out.

The log of the use of the system in support of the USAWC war games for 15-18 January 1962 is reproduced as Table 3. A total of 231 assessments were processed in 42 batches ranging from 1 to 43 cards each. For most of these batches, times in and out of the remote transmission and receiving station were recorded. The recorded elapsed times do not include keypunching and verification or tabulation time. No written record was kept of these times, but it is known that keypunching and verification times were generally about equal to the theoretical time of 1 card/minute. The reported elapsed times do include the two-way transmission time, the computing time, and time for all manual operations at the two fixed stations and within the mobile transmission and receiving station.

Table 3
Data Transmission Log, 15-18 January 1962

Date	Batch No.	Assessments in Batch	Time (hour) into Transceiver Station	Time (hour) out of Transceiver Station	Elapsed Time, min
15	1	1	nr ^a	nr	nr
	2	2	nr	nr	nr
	3	1	nr	nr	nr
	4	1	nr	nr	nr
16	5	1	nr	nr	nr
	6	14	nr	nr	nr
	7	1	nr	nr	nr
	8	1	nr	nr	nr
	9	5	nr	nr	nr
	10	5	nr	nr	nr
	11	17	nr	nr	nr
	12	4	nr	nr	nr
	13	3	1045	nr	nr
	14	4	1102	1105	3
	15	5	1113	1117	4
	16	1	1117	1123	6
	17	2	1129	1132	3
	18	9	1150	1159	9
	19	43	1223	1239	16
	20	1	1248	1254	6
	21	1	1322	1333	8
	22	1	1334	1340	6
	23	6	1415	1419	4
	24	5	1426	1431	5
	25	4	1434	1439	5
	26	4	1448	1455	7
	27	8	1456	1503	7
	28	1	1503	1508	5
	29	12	1537	1546	9
	30	11	1554	1601	7
	31	1	1616	1622	6
17	32	9	0851	0856	5
	33	4	0919	0926	7
	34	1	0944	0947	3
	35	16	0955	1001	6
	36	3	1002	1008	6
	37	1	1024	1030	6
	38	2	1121	—	—
	39	15	1355	1422	27 ^b
	40	3	1505	1514	9
18	41	1	—	—	—
	42	1	—	—	—

^aNot recorded.

^bComputer was down.

The shortest recorded time per batch was 3 minutes for batch 14, consisting of four assessments, and batch 17, consisting of two assessments; the longest recorded time was 27 minutes for batch 39 of 15 assessments, including some computer downtime. The largest batch processed, 19, consisted of 43 assessments and consumed 16 minutes of elapsed time. The average time for all recorded batches was 7 1/2-minutes. The average time for all recorded batches except 19 and 39, in which either there were large numbers of cards or the computer was down, was 6 minutes/batch.

To the best of the authors' recollections, the unrecorded batch times followed much the same pattern with the exception of several of the initial batches, which suffered additional delays because of procedural difficulties.

In addition to the assessment traffic described in Table 3, the transmission system was tested repeatedly with synthetic traffic between Carlisle and Gaithersburg throughout the period that it was in service (December - January). During this time, approximately 12,000 prepunched data cards were transmitted and checked for accuracy. The purpose of these preliminary tests was a) to aid in the engineering tests of the system, and b) to provide the operators with training and experience with the equipment in the environment of the actual test.

No cases of undetected errors in either transmission or computation were known.

COSTS

The out-of-pocket dollar costs of establishing the sample data-transmission system were very modest. The Signal Corps billed RAC between \$5000 and \$6000 for the installation and rental of all commercial terminal equipment and telephone lines and for the travel expenses of Signal Corps installation and maintenance personnel. In addition, RAC spent several hundred dollars directly for such items as wiring and carpentry materials, delivery costs of terminal equipment.

and movement of the RAC van by a commercial mover. Uncosted services consist of a) use of the RAC 1103A computer: b) use of certain government-furnished equipment, and c) technical personnel services primarily within Signal Corps and RAC.

SECTION IV

CONCLUSIONS

The two RAC activities in the development of techniques for man-machine war games led to these results.

- (a) The feasibility of remote computational support for war gaming was demonstrated.**
- (b) The mechanization of the air assessments enhanced the effectiveness of the USAWC war games. It not only saved time for the game participants, but also permitted the testing of alternative strategies, which would not otherwise have been possible.**
- (c) Without the requirement for the transmission of classified information, establishment of the data transmission system would have been simple and inexpensive. Although the requirement to transmit classified information did not appreciably increase the dollar cost, it created problems, e. g. , obtaining special cryptographic equipment, and establishing safe security procedures. All problems were solved for the exercise.**
- (d) For future Army users, the problem of cryptographic equipment availability should be simplified by an equipment acquisition program now under way. The problem of establishing safe security procedures probably cannot be alleviated in the near future.**
- (3) Although the test dealt with only one specific data transmission system and one specific application, the wide**

range of types of transmission channel and terminal equipment commercially available (at a price) and the flexibility of the special cryptographic equipment employed permit extrapolation of the feasibility of supporting other reasonably similar war gaming and scientific applications by computers at remote locations. In each case, however, a systems study should be conducted to confirm feasibility and to establish system specifications.

FORMAL STRUCTURES FOR INFORMATION SYSTEM DESIGN

FOREWORD

The author appreciates and wishes to thank C. B. McGuire of the University of California, Berkeley, for numerous ideas and comments; J. Petersen, H. W. Nelson, and W. Shelton of The RAND Corporation for their example of information system effectiveness; O. T. Gatto of the RAND Corporation for material on AUTOSATE, and S. L. Pollack of the RAND Corporation for material on decision tables.

FORMAL STRUCTURES FOR INFORMATION SYSTEM DESIGN

Richard L. Van Horn*

SECTION I

INTRODUCTION

Information system design has become a topic of prime importance. During this decade, for example, the United States plans to spend billions of dollars on an information system venture known as "Command and Control." In various ways these electronic data systems will provide military commanders with information about our forces, the enemy, and nature. The Command and Control effort encompasses immense problems, some common to most large information system development projects, and others unique. What jobs should these systems perform, for example, and during what periods? Some choices are: maintaining a high condition of readiness prior to commitment; making a commitment decision, identifying the enemy, targeting and notifying the troops to go at the time of commitment; and organizing second and follow-on strikes, stopping hostilities, negotiating, and organizing recovery in the post-commitment era. Furthermore, the same systems might play different roles in limited war or other stress situations. Each of these tasks implies different decision mechanics, information flows, and communication links. Further aspects are vulnerability, the role of contractors versus in-house capability, hardware requirements, realistic schedules, and endless others.

*The RAND Corporation, Santa Monica, California

Any views expressed in this paper are those of the author. They should not be interpreted as reflecting the views of the RAND Corporation or the official opinion or policy of any of its governmental or private research sponsors.

While specific hardware has been proposed to bolster present Command and Control structures, little has been done to design better information and decision systems. For that matter, the types of research that augment or complement design effort have been limited. The types of organizations that best serve particular commands, the types and amount of essential information, and the decision rules to use with the available information are but a few of the little-researched areas in Command and Control.

Current limitations in implementing these types of systems stem largely from the lack of design effort in information structures, decision-making, and management organization. Several of the largest Command and Control systems, for example, depend on the base as the first echelon for data input and response to command decisions; yet the process by which a base generates information and executes commands has received scant attention.

THE ESSENCE OF FORMAL STRUCTURES

One step toward defining and solving some of these problems is to develop more formal structures for information system design. Webster's Dictionary defines "formal" as "done in accordance with forms or rules; methodical." This definition points in the desired direction, but the goal here is more specific. In this context, formal structures for information system design involve, in addition, the following characteristics: a) formulation and investigation of alternatives, b) evaluation of cost and benefits associated with each alternative, and c) a mechanism for explicit communication of research.

The formulation and investigation of alternatives is an obvious but often neglected requirement. Many projects do look at alternative hardware configurations for doing a "given job," but such an examination is only part of the picture. A "given job" is seldom really "given." In many situations people operate reasonably well with no formal information system or with only a very simple

one, because they adapt their decision rules to the existing circumstances. The results from a well-designed, low information system are often surprisingly good compared with those of a high information one. In any case, alternatives are a prime requisite, both in the job to be done and how to do it.

The evaluation of costs and benefits is closely related to the question of alternatives. Attempts are made in many projects to evaluate hardware and development costs; but hardware costs, like hardware alternatives, are only part of the picture. A second basic design criterion is how much improvement in "management system performance" results from any particular information system. In other words, measures of both cost and effectiveness are needed. These ideas by themselves are neither novel or revealing; but it is significant that system performance can be, and in some areas has been, measured as a function of information.

A mechanism for explicit communication of research is certainly a major difficulty in information system design. People do design information systems that not only work, but often appear highly effective and efficient. For some reason, however, they cannot or do not communicate meaningfully what decisions they made and why. The "forms, rules, and methods" of information system design remain couched in vague and verbose generalities. Furthermore, when a top manager or review team attempts to evaluate a design effort, no one can really reproduce exactly what was done and why. The most explicit judgment we can pronounce on information system design is that as long as the current mode of communication between designers exists, we will not get very far.

Mathematics, of course, is an excellent means of explicit communication -- at least among mathematicians; but the problem is not solved by a decision to adopt mathematics as an official language as one might adopt ALGOL for

programming. The real task is to identify the key variables precisely, so that explicit statements about them – whether mathematical or otherwise – have a useful meaning

With more formal structures, one clearly hopes to do a more effective job or to do it more efficiently, or both. Information system design projects are characterized both by long elapsed times – several years to a decade – and by large workloads – hundreds of man-years. During the development cycle, new hardware emerges, policies and objectives change, and personnel changes. The large workload produces a large requirement for scarce people and makes control of the project difficult. As a consequence, a further goal is to reduce the elapsed time and total man-hours required for analysis, design, and implementation.

A great deal of relevant structure already exists. More than twenty years ago, Chester Barnard presented the view that the study of "organizations," long a subject of great interest to students of business, might profit from more concentration on decision-making aspects.^[1] Ten years later, stimulated by developments of Wald's decision theory and von Neumann's theory of games, Herbert Simon began to formalize the same notion.^[2] Bavelas and others began empirical studies^[3] of how small groups organize for decision-making purposes under rigid and explicit communication constraints and payoff functions. Shannon, although not talking directly about information in a management sense, clarified and formalized the notion of information.^[4] Wiener and others began to view organizations in terms of the servomechanism theory. This notion led in two directions: attempts to understand and model the brain (even to create intelligent machines), and attempts to model business organizations.^[5] Marschak and Radner used the essentials of decision theory to create a formal representation of a decision-making organization that they called "team theory."^[6] Although

none of these efforts provides a complete approach to information system design problems, they do suggest some ways of pinning down much of the unity that is recognized in this subject.

DEFINITION OF AN INFORMATION SYSTEM

The information system designer disposes of a definition for a Command and Control system by saying it is a "management information system" for the military. If one proceeds a step further, however, all agreement breaks down and endless argument goes on over what a management information system is and how it differs from a management control system, a data-processing system, and other variants. To work with formal structures, one needs a basic perspective. In most scientific fields, advances are accompanied by gradual changes in the meaning of important terms and by the development of a special vocabulary that – despite its barbaric appearance to outsiders – allows explicit communication among the initiated. Statistical decision theory can provide an explicit formal framework. The word "framework" is consciously chosen; a satisfactory fifteen-word definition, if one exists, is still to be found.

The types of problems under discussion first of all involve an organization. Webster defines "organize" and "organization" variously as "to arrange or constitute interdependent parts, each having a special function or relation with respect to the whole . . . any highly complex thing or structure with parts so integrated that their relation to one another is governed by their relationship to the whole." Within the organization, people, on the basis of available information about the state of the world and the effects of different actions, try to make decisions that will advance their interests. The available information may be more or less imperfect, and the interests of the people may or may not be similar. The actions taken plus the state of nature produce a payoff and put the organization into a new environment which, once again, is imperfectly made known to the participants for their next round of decisions. And so the process goes on.

Already the difficulties that face attempts to classify a particular system as an information system or a management control system are apparent. Most systems both produce information and involve decision rules. It is possible, however, to think of a hierarchy of systems.^[7] First-level systems contain decision rules only for data manipulation and in the short run, at least, the decisions are fixed. The Ballistic Missile Early Warning System (BMEWS), for example, supplies data under a fixed and predetermined program. Most military and commercial data processing systems are of this general type. Second-level systems allow the user to take short-run actions that affect the supply of data. In Electro-magnetic Intelligence System (ELMINT), for example, the rate and type of data supplied are subject to control. Since the user tends to get only the data he wants in a second-level system, it might be classified as a system for "producing information." Third-level systems supply data and allow the user to take actions that affect both the supply of data and the state of the world. These systems thus contain three levels of decision mechanisms: a) a mechanism for data manipulation; b) a mechanism to take actions that change the data-manipulation rules, and c) a mechanism to take actions that change the state of the real world. The SAC Control System is one example of a third-level system -- a system for "taking action."

The subsequent sections of this paper discuss formal analysis techniques for information requirements, information flows, and system development.

SECTION II

EVALUATION OF INFORMATION AND DECISION ALTERNATIVES

The notions of payoff, information, and decision rules can be stated more explicitly.^[8] Consider a single decision-maker faced with uncertainty. Let "x" denote the state of Nature. The payoff to the decision-maker, if he chooses a particular action or strategy "b" when the state of Nature is "x," is $u(b, x)$. If he follows Savage's decision criterion, then he chooses that action "b" which maximizes the expected value of the payoff, $EU(b, x)$.^{*}

In order to look at the role of information in more detail, suppose there are several forecasters or information systems, $\eta_1, \eta_2, \eta_3, \dots$, whose services the decision maker can purchase. Each one gives forecasts or signals "y" which depend on "x" in a known way: $\eta(x) = y$. If two or more different states, x_1, x_2, \dots , of the environment yield the same information to the decision-maker, i. e., if $\eta(x_1) = \eta(x_2) = y$, then he is uncertain about which state is the true one. Inaccuracies and mistakes in forecasts are portrayed in this way. Clearly, the different forecasters will have different characteristics; one may be unable to distinguish between x_1 and x_2 ; another between x_2 and x_3 . Thus, for two information systems η_1 and η_2 , we might have

$$\eta_1(x_1) = \eta_1(x_2) \neq \eta_1(x_3)$$

and

$$\eta_2(x_1) \neq \eta_2(x_2) = \eta_2(x_3).$$

^{*}Savage advances a descriptive theory of decision-making which argues that a rational man's decisions will perhaps (not always without some mathematical advice) exhibit a consistency that implies the existence of a von Neumann-Morgenstern utility function and a subjective probability distribution over the set of states of nature.^[9]

The functions, η_1, η_2 , each of which divides the set of possible states of the environment into distinguishable subsets, are alternative information structures. The real interpretation of these structures will depend on the context. Observation facilities, communication systems, computing and display hardware, and many other physical and human mechanisms will ultimately determine what alternative information structures are available to decision-makers.

Given some information structure " η ," the decision-maker now chooses a rule " α " which tells him what action " a " to employ in response to a given signal " y ," or alternately, $\alpha(y) = a$. Returning to the earlier formulation, an action " b " now consists of two parts: choice of an information structure " η ," and choice of a response rule " α ." Thus $b = (\eta, \alpha)$. The expected value of the payoff can now be formulated as

$$Eu(\eta, \alpha, x)$$

where

" η " is the information structure, " α " the decision rule, and " x " the state of the real world.

A MISSILE SYSTEM EXAMPLE

A simple example will help to clarify this approach. Assume that a missile complex contains four hardened and dispersed launch sites, each of which has one missile. This complex has four targets, A, B, C, and D. The value of successfully launching a missile at A is 400 points; B, 300; C, 200; and D, 100. When everything works perfectly, there is no problem. If a launch signal is received, one missile is sent at each target and a payoff of 1000 results.

If, on the other hand, missile unreliability or enemy action reduces the probability of a successful launch at any site to 0.5, then some questions arise.

The missile aimed at A – the highest priority target – may fail to get off, while the missile aimed at D – the lowest priority target – is successful. Other similar outcomes that appear undesirable may occur. One solution is to construct a hardened "Command and Control" system to connect the sites together; however, such a system is costly and the question arises, "What is it worth?"

With a "perfect" Command and Control system, all the sites can communicate with each other. The information structure contains full information; each site knows the condition of all others. The best decision rule for this information structure is: "Attempt to fire at A until a successful launch is made. If a missile remains, fire at B until successful, then at C, and finally at D in similar fashion." If three sites are disabled, the remaining one fires at target A, while if two remain in ready condition, they fire at A and B. The different real-world states or outcomes that are possible, and their probabilities of occurrence, are shown in Table 1.

Table 1
Probabilities of Real-World States

Real-World Outcome	x_1	x_2	x_3	x_4	x_5
Missiles Launched	4	3	2	1	0
Targets Covered	A, B, C, D	A, B, C	A, B	A	None
Probability of Occurrence	1/ 16	1/ 4	3/ 8	1/ 4	1/ 16

The expected payoff from this scheme is

$$\frac{1}{16} (1000) + \frac{1}{4} (900) + \frac{3}{8} (700) + \frac{1}{4} (400) + 0 = 650.$$

The missile aimed at A – the highest priority target – may fail to get off, while the missile aimed at D – the lowest priority target – is successful. Other similar outcomes that appear undesirable may occur. One solution is to construct a hardened "Command and Control" system to connect the sites together; however, such a system is costly and the question arises, "What is it worth?"

With a "perfect" Command and Control system, all the sites can communicate with each other. The information structure contains full information; each site knows the condition of all others. The best decision rule for this information structure is: "Attempt to fire at A until a successful launch is made. If a missile remains, fire at B until successful, then at C, and finally at D in similar fashion." If three sites are disabled, the remaining one fires at target A, while if two remain in ready condition, they fire at A and B. The different real-world states or outcomes that are possible, and their probabilities of occurrence, are shown in Table 1.

Table 1

Probabilities of Real-World States

Real-World Outcome	x_1	x_2	x_3	x_4	x_5
Missiles Launched	4	3	2	1	0
Targets Covered	A, B, C, D	A, B, C	A, B	A	None
Probability of Occurrence	1/ 16	1/ 4	3/ 8	1/ 4	1/ 16

The expected payoff from this scheme is

$$\frac{1}{16} (1000) + \frac{1}{4} (900) + \frac{3}{8} (700) + \frac{1}{4} (400) + 0 = 650.$$

With good decision rules and a "perfect" Command and Control system, this missile complex still produce a payoff of 650, even though the probability of success for each site is only 0.5.

A second problem, however, is what can be done without a "perfect" Command and Control system. In other words, what are the alternatives? One alternative is an information structure that provides to each site no information on the current status of other sites. In this event, the problem reduces to looking for good decision rules, or in this case, how to assign targets to sites. If A is assigned to Site 1, B to Site 2, C to Site 3 and D to Site 4, then the payoff is:

$$\frac{1}{2} (400) + \frac{1}{2} (300) + \frac{1}{2} (200) + \frac{1}{2} (100) = 500.$$

It is obvious, at this point, that a perfect information system is not worth the difference between no payoff and one equal to 650. At best, it is worth the difference between a payoff of 500 and one of 650. The analysis should not end here. It is possible, perhaps, to find decision rules better adapted to the no-information case. Some possibilities are shown in Table 2.

Table 2
Probable Target Coverage

Decision Rule	Target Assignment Site				Expected Payoff
	1	2	3	4	
$\alpha 1$	A	B	C	D	500
$\alpha 2$	A	A	A	A	375
$\alpha 3$	A	A	B	C	550
$\alpha 4$	A	A	B	B	525

Other possibilities are feasible, but rule 3, which shows the best payoff in Table 2, is actually the best over-all rule. In the Table, the maximum worth of a "perfect information system" is the difference between a payoff or target coverage of 650 and one of 550.

Other "partial" information structures exist in addition to the full and no-information ones. [10] For example, one might connect Site 1 with Site 2 and Site 3 with Site 4 by a hardened communications system. The search then begins for good decision rules under the circumstances. Or, all sites might be connected by regular land lines, a vulnerable but cheap system. Each of these alternatives will yield its own cost and payoff, and can be compared to the others.

The preceding example, although highly simplified, illustrates the notions of information structures, decision rules, and payoff. It is particularly interesting to note, first of all, that the null or no-formal-information system case may not be a hopeless situation. Careful choice and coordination of decision policies sometimes produce an acceptable result. Second, with a given information structure, examination of alternative decision rules is important. Third, the increase in performance between the null case and full-information case is the maximum worth of an information system. Finally, many partial information structures exist and require consideration. In practice, this type of analysis is difficult and contains many complicating factors, but in some situations it is feasible. The next section describes one such project.

EVALUATION OF INVENTORY SYSTEMS

An effective information system is a vital prerequisite to managing and operating a military inventory system. Figure 1 shows the basic flows in one such system.

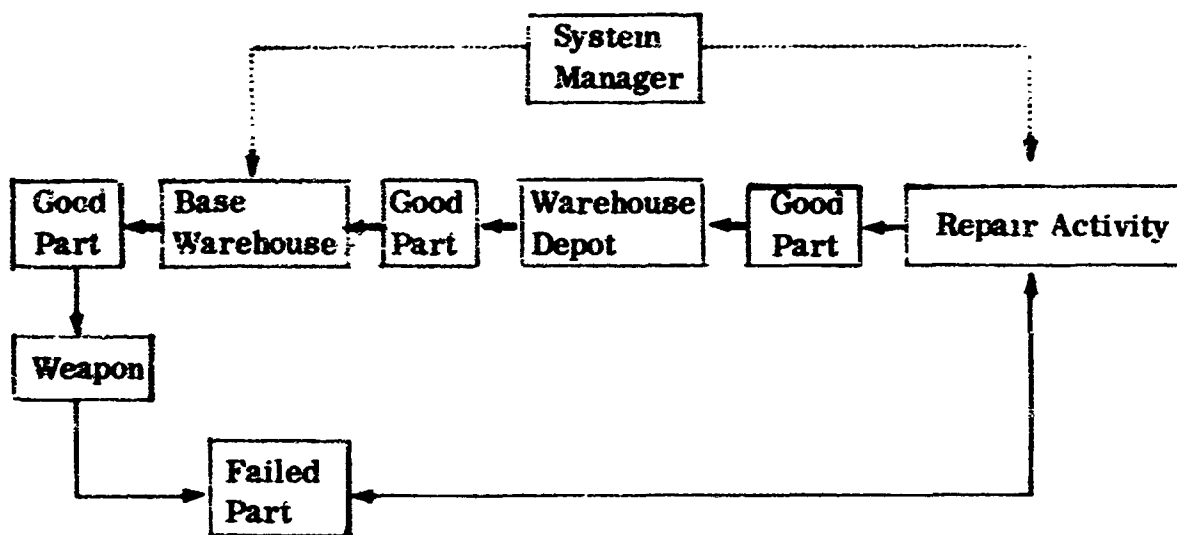


Fig. 1. Basic Flows in a Military Inventory System.

In a simplified form, the system operates as follows. When a part in a weapon fails, it is replaced with a good part from the base warehouse. The failed part is returned to a repair activity. When a base warehouse reaches its reorder point for a particular item, a supply of good parts is sent from a depot warehouse. Meanwhile, the failed parts are repaired at the repair activity and returned to the depot warehouse. The actual operation of the system is, of course, much more complex with many bases, line items of inventory, warehouses, and repair activities, plus many other possible paths through the system. [11]

To examine information system effectiveness requires a) selection of a measure of effectiveness; b) alternate information structure and the decision rules, and c) an evaluation of the payoff or effectiveness for each set of decision rules and information structures. As described earlier, the designer chooses an information structure and decision rules which, when played against real-world outcomes, produce some payoff. In this case, stockout-weeks was chosen as the measure of system performance or effectiveness. If a weapon

requires a good part and none is available, a stockout occurs until a good part is available. Under this criterion, an information system that results in fewer stockout-weeks than another is regarded as better or more effective.

INFORMATION STRUCTURES AND DECISION RULES

Information and decisions enter the process in the following fashion. The program for the repair activity depends on the relative need for different items; critical items are repaired first. Under current procedures, approximately eight weeks are required to a) collect information from bases and warehouses about parts on hand; b) process this information, and c) prepare a repair schedule. As a result, repair action is started on the basis of a situation that existed eight weeks in the past. In similar fashion, an average of twelve days is required from the time a base submits a request for parts to the time it receives them. Finally, parts are moved from one base to another only after a stockout occurs.

Experience indicates that this system results in a relatively large number of stockouts; consequently, an improvement in effectiveness is a major objective. Some possible choices are to:

- (a) provide hardware and procedures to reduce the repair action delay from eight weeks to a smaller figure, perhaps four weeks or one week – an information structure change,
- (b) provide hardware and procedures to reduce the distribution cycle from twelve days to a smaller figure – also an information structure change, and
- (c) change the decision rules to allow moving parts from one base to another whenever a critical level is reached and the depot warehouse has no parts.

Although the last choice concerns decision rules,
note that it also involves the information structure.
The ability to move parts from one base to another
implies information about parts on hand at each base.

SYSTEM PERFORMANCE

All of the above approaches appear reasonable and intuitively desirable in that they might increase effectiveness. The standard approach, at this point, is to begin assembling the hardware for a decision and information system with some "reasonable" capability. Some very important questions, however, remain unanswered. Will any of the above alternatives actually improve effectiveness - decrease stockout-weeks? If so, how much? Which change produces the greatest improvement? The answers to these questions, if available, provide a rational basis for design.

Table 3 shows the performance measures associated with some specific alternatives. The numbers were generated by use of a complex computer simulation model based on real-world observation of policies, procedures, and performance. Case 1 represents the existing system. All other choices did indeed improve performance, but by varying amounts.

SELECTION OF AN ALTERNATIVE

The next step is to develop in detail the hardware and procedures required to implement the cases that provide acceptable performance, and to estimate costs. The requirements to implement alternatives will often vary greatly. It may be possible, for example, to obtain Case 2 at low additional cost by minor modifications to the existing system. Case 9, on the other hand, may require large-scale processing equipment, high-speed communication facilities, extensive system redesign, and a several-year development cycle.

Table 3

Inventory System Performance in Cumulative Stockout-Weeks*

Repair Response: Length of Management Response Cycle	Distribution Response in Stockout-Weeks		
	Average Resupply Cycle 12 Days; Base Redistribution for Stockouts Only	Average Resupply Cycle 4 Days; Base Redistribution for Stockouts Only	Average Resupply Cycle 4 days; Base Redistribution Whenever Critical Levels are Reached
8 weeks	Case 1 3364	Case 4 1825	Case 7 1656
4 weeks	Case 2 2749	Case 5 1130	Case 8 1035
1 week	Case 3 2491	Case 6 886	Case 9 706

* From an analysis by H. W. Nelson and W. Shelton of data in Reference 11.

A particular information structure/decision combination may itself be implemented in many ways. The base redistribution feature of Cases 7, 8, and 9 might, for example, be obtained by establishing a processing center that maintains current records of the inventory at all bases. When a critical situation develops, a central manager directs a transfer from one base to another. The same feature could also be incorporated in a decentralized system by proper coordination. For example, when a base reaches the critical level, the base manager queries other bases until he finds one with adequate stock. The base with adequate stock then ships to the short base. Careful selection of the implementation mode may in itself greatly reduce the cost of a particular information structure. The analysis and design of systems to implement information structures is discussed in more detail in the following sections.

When measures for cost and effectiveness are available, final selection of a design can be made. For example, if it costs the same to get from the Case 1 to either the Case 2 or Case 4 system, then the Case 4 system is clearly

preferable to Case 2 since its performance is much better. After reviewing cost and effectiveness, the designer may conclude that none of the possible choices are desirable – all cost too much for the improvement in effectiveness. At this point, the search starts again for a method that has a better cost-effectiveness measure.

This example is highly simplified, of course. The actual study includes consideration of many more factors. Certain problems, however, exist even here. Is stockout-weeks a valid measure of performance? Were certain features that might improve performance or reduce costs overlooked? Did the simulation model produce reliable numbers? These and other questions remain to be answered by management judgment and experience. The only purpose here is to illustrate that explicit measures of effectiveness for an information system are possible and essential to rational design.

SECTION III

INFORMATION FLOW ANALYSIS

Once the desired information structure and decision rules are established, the job of developing and implementing a large information system is still difficult and hazardous. One common task in system design is the analysis of information flows in a specified system. Such a task may arise as a desire to study either an existing system or a proposed new one. Flow charts are the typical approach; however, this approach has numerous drawbacks. It requires large expenditures of time by experienced analysts and is prone to error at many points. If the object is merely to automate the specified system, then detailed flow charts may be the most reasonable way to proceed.

Often, however, there are other objectives. The existing system may appear undesirable, and the object of analysis is to aid in the design of a new system. The current flow paths, inputs, output, and files of the existing system can provide a checklist for the new system. The analysis of a proposed system looks to see if it is logically consistent and does the job it set out to do. Several desired characteristics of a technique for analysis of both proposed and existing systems are:

- (a) a minimum of routine data-manipulation by the analyst,
- (b) built-in checking features for logical flows,
- (c) quantitative outputs that characterize the system under study, and
- (d) a framework that aids in estimating the cost of the system.

Conventional flow charts possess none of these characteristics and are of little help. The analyst simply doesn't know what to do with the huge mass of paper that faces him.

AUTOSATE – Automated System Analysis Technique – is an attempt to develop procedures that meet the above criteria. [12] AUTOSATE consists of three processes: 1) input collection; b) input processing and report preparation and c) output use. The second process, which in conventional flow charting consumes most of the analyst's time, is entirely mechanized in AUTOSATE. Furthermore, input collection is simplified and standardized so that non-analyst personnel can be trained to perform it. Finally, each report produced by AUTOSATE has its own specific uses.

To apply AUTOSATE, the organization is first divided into "stations." A station is a group of people, functions, or hardware – a subunit of the organization – treated by AUTOSATE as a single node in a data flow network. In a detailed study, each person might be a station; in a more aggregate study, a station might be a department. Data in the form of messages originate at and flow between stations. Files exist at stations. Stations, messages, and files are connected together in event chains. For example, the detection of a missile is an event that might generate a message at a station. This message then goes to other stations and in turn may result in the generation of hundreds of other messages. The whole sequence of actions that result are the event chain for the event – missile detection. The key pieces of AUTOSATE are, therefore, messages, stations, flows, files, and event chains.

DATA COLLECTION

To conduct the analysis, an interviewer goes to each station and documents on a Message Specification Sheet all the messages that come into the station, are generated within the station, and leave the station. Files are also documented.

There is no need to segment the system into applications or other functional partitions. All messages and files – formal and informal and regardless of nature or content – are documented as part of a particular station. There is, furthermore, no need for the interviewer to connect up or trace flows; the machine processing will build event chains at a later step.

The Message Specification Sheet formalizes the data-gathering process and is the foundation upon which the majority of future analyses are based. Since it is an important document, details of its contents are described in this section and a sample shown in Fig. 2.

Form Number is the given Air Force or Command Number for the form, if one is involved. If a message has no number, as it often happens if it is a local form or verbal message, one is assigned during the editing phase. Station refers to the particular suborganization in which interrogations take place. Identifier classifies a message by its type: input, output, and so forth. Event (or sequence) describes the action or event that caused the message to arrive at, generate in, or leave this station. Source identifies the station from which the message was received.

Processing actions are of two types. One is an action in which a message acts upon or with another message or file. In this case, the specific message or file acted upon is noted in the space provided in Section A of the message specification sheet. The second type of processing action describes how a message entering one station affects another. The affected station is designated in the space provided in Section B of the form. For example, if aircraft are scrambled upon receipt of an alert message, the action taken is "scramble" and the station affected is "aircraft."

Frequency denotes the general processing period associated with the information. Special time requirements identify the special processing required for

CARD CODE (1)		FORM NO. (2-8)		FORM TITLE (9-24)	
/		L C 200 alpha numeric		PREPARED NOTICE	

STATION (25-28)	
activity name	3000
code	

EVENT or SEQUENCE (30-49)	
Code	IFINLABORSTDS

IDENTITY (29)	
Output	O 1
File/Output	FO 2
Report	R 3
Input	I 4
Input/Output	IO 5
Input File	IF 6
In/File/Out	IX 7
File	F 8

PHYSICAL FORM (50) all codes except 8		FILES - code 8	
Document (manual)	DM 1	Drawer	1
Document (typed)	DT 2	Folder	2
Document (process)	DP 3	Chart	3
Punch Card	PC 4	Status	4
Telephone Call	TC 5	Sheet	5
Teletype Message	TM 6	Ledger	6
Radio Message	RM 7	Kardex Cards	7
Verbal Message	VM 8	3 x 5 Cards	8
Paper Tape	PT 9	Tag	9
Magnetic Tape	MT 0		

SOURCE (51-54)	
activity name	3000
code	

PROCESSING ACTION (55-56)	
Section A: actions affecting forms	
Section B: actions resulting in other actions	
Section A	10 Operate 11 update 12 extract 13 pull 14 post 15 prepare 16 compare with 17 reference 18 complete 19 top
Section B	20 Sign 30 Transport 40 Delay 50 Store 60 Inspect

FREQUENCY (64)	
Hourly	H
Daily	D
Weekly	W
Semi-monthly	S
Bi-weekly	B
Monthly	M
Quarterly	Q
Semi-annually	E
Annually	A
As Required	R

SPECIAL TIME REQ'MTS (65-66)	
None	blank
Immediate	IM
Hour	
Day	-
Month	+

VOLUME (67-70)	
34	

DISPOSITION (71-75)	
STATION NO. 71-74	
7000	
Hand carry	1
Telephone	2
Verbal	3
Radio	4
Regular Mail	5
Airmail	6
Teletype	7

DISPOSITION (76-80)	
STATION NO. 76-79	
Hand carry	1
Telephone	2
Verbal	3
Radio	4
Regular Mail	5
Airmail	6
Teletype	7

FORM NO. (57-63)	
FIL 205	

STATION NO. (57-63)	

Fig. 2. Message Specification Sheet

certain information such as immediate processing, or processing by a specified time. Volume represents the number of messages processed during the frequency period. Disposition designates the station to which the message is next sent after processing at a particular station is completed.

The interview phase for a station is complete once all the messages processed by it have been documented. A single interviewer might proceed from station to station until all are covered. In a large system, however, the long time required for a serial approach is undesirable. Since data collection at each station is independent and a standard, explicit collection process is used, as many interviewers as desired can operate simultaneously.

EDITING AND PREPROCESSING

Following the interview, the specification sheets are forwarded to a control point for editing and coding. The editing and coding phases of the process are the "clean-up" points prior to releasing data to a key punch operator. Specification sheets are checked for completeness and correctness of data content. Station, source, and disposition codes are obtained from a master listing and inserted into the appropriate areas on each of the specification sheets. Numbers are assigned to local forms and files. The event description section of the document specification sheet is reviewed and abbreviated, if necessary, to fit the coding area. At this point, specification sheets are ready for the preprocessing phase.

The preprocessing phase is the last point in the over-all process where the human manipulates raw data. Beyond this point, the processor manipulates the data in a number of different ways to provide the analyst with various outputs. Preprocessing includes the transcription of inputs to machine media and preparation of information tables to facilitate subsequent computer processing. At the end of preprocessing, the machine run occurs, reports are produced, and the analyst can begin his examination.

ANALYSIS WITH AUTOSATE

With AUTOSATE, system analysis provides an explicit and quantitative understanding of a specified information system in terms of the characteristics of each station in the system and the relationships of stations with one another.

Stations are characterized by type of data processed, the frequency of processing, volume processed, and any special processing required. Depending upon the station's classification as satellite, control, or storage point, the systems analyst correspondingly will concentrate upon terminal, computer, display, or storage requirements and capability. Knowing the station's data processing characteristics helps the analyst measure the relative importance of the different stations in the organization insofar as data processing needs pertain. The relationships of stations with one another give the analyst a measure of logical flows -- correctness, completeness, and simplicity -- and the potential for restructuring -- integration, centralization, or simply different flows.

Determining station characteristics and relationships by conventional system analysis methods is extremely complex and tedious. Doing so manually takes up the system analyst's time, which can be better applied to analysis than to data manipulation. To help the analyst in his job, AUTOSATE has two major outputs: event-chain flow charts and station characteristics. These reports are produced entirely by the processor. The analyst concentrates on their use, not their preparation.

Event chain flow charts are a means of relating data processing activities between stations. The event chain flow chart emphasizes the event that creates a message or activity at a particular station. Consider, for example, the report of a malfunction at a debriefing area -- a station. The event is "debriefing" and the document generated is a debriefing form that details the malfunction.

The form is forwarded to the maintenance control center -- another station -- where the event "receipt of debriefing form" causes preparation, posting, or referencing of additional records and documents. The maintenance control center requests dispatch of a specialist to make the repair, thereby creating the third link in the event chain, since a "request for dispatch" is the event which causes the mechanic in the maintenance shop -- another station -- to undertake the repair action. Note that an output at one station becomes an event that triggers action at a subsequent station. Failure to detect or record any link in this chain results in an incomplete chain -- an output with no input, or input with no output. At the end of processing, two types of flow charts are created: those showing complete event chains, and those showing incomplete ones. Incomplete chains may occur because the interviewer failed to collect complete and accurate data or because a logical error or inconsistency exists in the data system. AUTOSATE detects the error and calls it to the attention of the analyst, but the analyst must identify the type and cause.

Figure 3 shows an event chain flow chart from an analysis of a proposed Air Force Depot Management System. This chart, except for the straight lines, was put together and printed entirely by the processor. The event that starts this chain -- Chain No. 24 -- has a sequence number of zero and is a requirement to "PREPARE (a) REPORT" on skill control. The first station in the chain is DATA SERVICES, and a local form numbered E400 (LOCE400) is involved. The identity (ID) of the operation is to prepare an output from a file (FO). The DP code in the column headed FM indicates that the form is prepared by machine processing; the Q in column F indicates this event occurs quarterly. No special time requirement (SPEC TIME) exists. Volume (VOL) indicates that 29 different Skill Reports are prepared each quarter. PROCESSING contains an O for operation. The specific operation was EXTRACT (from a file) as shown under ACTION VERB and the file affected was FIL 52. The report produced was

SEQUENCE NUMBER	EVENT	STATION	DOCUMENT NUMBER	ID	FM	F	SPEC TIME	VOL	PROCESSING					ACTION VERB	DOC/ STAT	DISPOSITION		SEQUENCE NUMBER
									U	T	D	S	I			HOW	STATION	
24 1000000.	PREPARE REPORT	DATA SERV	LOCE400	FO	DP	Q		0029	X					XTRACT	FIL 52	HNDCY	PLANNING	24 1000000.
24 2000000.	REG DISTR/SKIL SHORT	PLANNING	LOCE400	IF	DP	Q		0029					X	INSPEC	LOC220	HNDCY	PLANNING	24 2000000.
24 2100000.	REVIEW SKILL SHORTGE	PLANNING	LOCE400	IX	DP	Q		0001	X					PREPAR	LOC220	HNDCY	MANPOWER	24 2100000.
24 2200000.	RETAINING NOT POSSBL	MANPOWER	LOCF220	FO	DT	Q			X					PREPAR	LOC210	HNDCY	DESTROY	24 2200000.
24 2210000.	REVIEW REQUEST+ANSWR	MANPOWER	LOCF210	IX	DT	Q								PREPAR	LOC210	HNDCY	PLANNING	24 2210000.
24 2220000.	REPLY TO REQUEST	PLANNING	LOCF210	O	DT	Q			X					PREPAR	LOC230	HNDCY	PLANNING	24 2220000.
24 2221000.	REVIEW/IF APPROVAL	PLANNING	LOCF210	IF	DT	Q								PREPAR	LOC210	HNDCY	PERSONNEL	24 2221000.
24 2222000.	IF ADDL PERS AUTH	PERSONNEL	LOCF230	O	DT	Q			X					PREPAR	LOC210	HNDCY	PERSONNEL	24 2222000.
24 2223000.	IF CANNOT HIRE SKILL	PERSONNEL	LOCF210	IF	DT	Q								STORE	FIL 217	HNDCY	PLANNING	24 2223000.
24 2224000.	INFORMING SHOPS	PLANNING	LOCF210	O	DT	Q							X	STORE	FIL 217	HNDCY	PLANNING	24 2224000.
24 2225000.	IF NO AUTHORIZATION	PERSONNEL	LOCF230	IX	DT	Q							X	PREPAR	LOC210	HNDCY	PERSONNEL	24 2225000.
24 2226000.	DETERMINE NEXT ACT	PERSONNEL	LOCF210	IF	DT	Q			X					STORE	LOC210	HNDCY	PLANNING	24 2226000.
24 2227000.	IF HIRE NEW SKILLS	PERSONNEL	LOCF210	IF	DT	Q								STORE	FIL 218	HNDCY	PLANNING	24 2227000.
24 2228000.	NOTIFY MAINTENANCE	PERSONNEL	LOCF210	O	DT	Q							X	PREPAR	LOC210	HNDCY	DESTROY	24 2228000.
24 2229000.	RECEIPT OF PERS ACT	PERSONNEL	LOCF210	IF	DT	Q								STORE	LOC210	HNDCY	PLANNING	24 2229000.
24 2230000.	IF NEW HIRES REQ TRG	PERSONNEL	LOCF210	IX	DT	Q			X					STORE	LOC210	HNDCY	PLANNING	24 2230000.
24 2231000.	NOTIFY MICE OF ACT	PERSONNEL	LOCF210	O	DT	Q								STORE	FIL 218	HNDCY	PLANNING	24 2231000.
24 2232000.	NEW HIRE TRG NOTICE	PLANNING	LOCF240	IF	DT	Q							X	STORE	FIL 218	HNDCY	PLANNING	24 2232000.

END OF THIS CHAIN

Fig. 3. Event Chain Flow Chart

hand-carried (HNDCY) to the station called PLANNING. The final column shows the sequence number of the next step in the chain which may be either the next item in the main chain or the first step in a sub-segment of the main chain. Sequence numbers therefore provide the equivalent of branching in a conventional flow chart. Subsequent entries in Chain 24 show preparation of other documents, storing of data in files, and so forth.

Event-chain preparation exhibits many of the desirable features mentioned earlier. Except for input collection, no manual effort is involved. Automatic checking for consistency and completeness is incorporated.

The structure of the output is a feature of particular interest. With conventional flow charts, structure is predetermined. The system is partitioned into applications -- groups of individual tasks that the analyst thinks or hopes are related. This prestructuring tends to bias subsequent design. Furthermore, data collection often presents problems because people in the system do not necessarily think of themselves as part of a particular "application." With AUTOSATE, structure is a result of analysis; the event chain procedures group together all of the individual actions that comprise a complete chain. These resulting event chains appear to offer an interesting starting point for restructuring the information system.

The second major AUTOSATE report is a straightforward listing of station characteristics. It includes the average number of each type of event that occurs at each station in a month -- outputs, inputs, input/file, etc. Each type is shown as a percentage of total activity at a station; and total activity at each station is shown as a percentage of total system activity. For each station, this report also shows the identities and volumes of the three other stations that have the highest data flow volume. This report, as described earlier, is used to identify key stations, the nature of each station -- input generator, storage point, etc.,

and stations with strong relationships – large data flows between them. The characteristic report quantifies the behavior of all the stations in the system. Since this report is a numerical summary of activity at each station, it is an excellent starting point for costing a system.

AUTOSATE, in summary, is an attempt to develop a more formal analysis structure, and has the following properties:

- (a) Event chain flow charts do not require the artificial partitioning of systems by application.
- (b) The standard data gathering process and station independence during the data collection permit simultaneous interviewing with any size group of interviewers.
- (c) The standard, clearly defined interview form makes it easier to train people in its use and results in fewer data gathering errors.
- (d) The event chain flow chart process imposes a system of internal control by calling attention to incomplete chains.
- (e) The mechanized data compilation process frees the analyst for more useful tasks commensurate with his abilities and reduces both man-hours and elapsed time.
- (f) Computer processing yields products not normally available by manual methods.
- (g) Explicit directions for the use of each output prevent overlooking more important aspects of the system.
- (h) Quantitative measures for each station provide a starting point for costing a system.

Like most development efforts, AUTOSATE has many deficiencies. For example, it measures volume as number of messages. Characters transmitted or work required at each station might be better measures. Only information flows, not the worth of a job, are dealt with. Without a separate analysis of information requirements, one could easily design excellent data flows to perform a nonexistent or worthless job. Despite these and other problems, however, AUTOSATE has proved very useful to date.

SECTION IV

SYSTEM DEVELOPMENT

Although this paper is primarily concerned with analysis of information requirements and information flows, similar techniques are relevant in the area of information system development. Development, as viewed here, is the process of going from a detailed plan to an operating system and includes programming, testing, and implementing a plan.

FORMAL STRUCTURES FOR PROGRAMMING

The programming of data processors consumes large amounts of dollars and manpower. In addition to the basic requirement, the user often has to repeat much of the programming when a programmer leaves in the middle of a job, a policy change occurs, or equipment is replaced.

A more formal technique for programming should have the following features. First, it should provide an orderly method of documentation. A good technique will make it relatively simple to document a job. Second, the technique should be independent of the processing hardware. The statement of the problem will then retain its usefulness, even if the processing method changes. Third, it should provide the system designer with flexibility to change portions of his analysis. Fourth, it is highly desirable to have a format that helps the analyst to visualize complex relationships. In large information systems, there are numerous complex relationships among the data. They are extremely difficult to visualize and analyze when they are described in English or algebra. Fifth, the techniques should facilitate review of a system description for omissions and inconsistencies.

One technique that does possess these characteristics, to some degree, is the use of a tabular or decision table format for programming.^[13] Table 4

shows a program written in both an English form and tabular form language. In the table, all entries above the horizontal double line are conditions; all below are actions. The top horizontal line is read as "if." All other single horizontal lines are read as "and." The horizontal double line is read as "then." Each rule in this decision table is read down. The "Y" says the condition must be satisfied; the "N" says the condition must not be satisfied; a blank or - means that this condition need not be considered in this rule. The X says, "Execute the action described"; a blank or - says, "Do not execute the rule described." Rule 1 in the table corresponds to the first paragraph and Rule 2 to the second paragraph. To illustrate the use of the table, Rule 1 means

```

IF TRANS-CODE = LOCAL REQUEST
and TRANS-CODE = WHSE-REFUSAL
and EXCHANGE-CODE = NO SUBSTITUTE
and REQUEST-ACCOUNT = 01
then Compute WHSE-REFUSUAL-AMOUNT
and Add WRA to ITEM-BALANCE
and Subtract ISSUE from S-144 REPORT
and GO TO TABLE 12.

```

It is much easier to detect errors in tabular form than in English format. All values used in comparison appear on the one line, not in different paragraphs. This makes it much easier to spot inconsistencies and other errors in these values. Having the conditions laid out in this tabular form enables the system designer to make more accurate determinations if he has considered all the possible combinations of conditions that might occur. He knows, for example, that if there are five conditions and each can be satisfied or not satisfied, then there is a total of 32 different rules he might form. The explicit format of the table also simplifies communication between programmers working on a job.

Table 4

English and Tabular Formats

	RULE 1	RULE 2		RULE 30
TRANS-CODE = LOCAL REQUEST	Y	Y		—
TRANS-CODE = WHSE-REFUSAL	Y	N		—
EXCHANGE-CODE = NO-SUBSTITUTE	Y	—		—
REQUEST-ACCOUNT EQUALS	01	"WSM"		—
ITEM-BALANCE REQUEST-AMT	—	Y		—
TRANS-CODE = OFF-BASE-REQUEST	—	—		Y
COMPUTE WHSE-REFUSAL-AMOUNT	X	—		—
ADD WRA TO ITEM-BALANCE	X	—		—
SUBTRACT ISSUE FROM S-144 REPORT	X	—		—
GO TO TABLE	12	15		60

IF TRANSACTION-CODE EQUALS LOCAL-REQUEST AND EQUALS WAREHOUSE-REFUSAL AND EXCHANGE-CODE EQUALS NO-SUBSTITUTE AND REQUEST-ACCOUNT EQUALS 01 THEN COMPUTE WAREHOUSE-REFUSAL-AMOUNT AND ADD WAREHOUSE-REFUSAL-AMOUNT TO ITEM-BALANCE AND SUBTRACT ISSUE FROM S-144-REPORT AND GO TO TABLE 10.

IF TRANSACTION-CODE EQUALS LOCAL-REQUEST AND TRANSACTION-CODE IS NOT EQUAL TO WAREHOUSE-REFUSAL AND REQUEST-ACCOUNT EQUALS WSM AND ITEM BALANCE IS GREATER THAN OR EQUAL TO REQUEST-AMOUNT THEN GO TO TABLE 11.

SYSTEM TESTING

As mentioned earlier, testing a completed system is a difficult task in any information system design project. Even with the best programming techniques, test decks, and all the standard checks, serious errors still show up during field operation of major projects. This type of error is especially undesirable in a system of the Command and Control variety since errors can have extremely serious consequences. In any case, errors of this type can require changes that are both expensive and time-consuming. Part of the problem is that testing starts too late. The ideal time to start testing is in the early development stage of a project before any programming is done. [14]

The first question to be asked about a new system is, "Will the customer understand and use the outputs?" The first step, therefore, is to mock up the output processes – hardware, forms, procedures, and decision rules – and play through the simulated situations that the system must deal with. Unclear decision rules, missing data, and numerous other problems become apparent during this step. The second test step is to mock up the various input processes and again play through the input situation. This step is particularly useful for determining the error checks on input that should be incorporated into the system. Note that both the first and second steps of testing can and should occur at a very early point in system development.

Once the basic processing procedures are laid out, the third test step can begin. It consists of reverse automation – using people to do the job of the computer. Various people, for example, can be assigned the tasks of replicating files while others replicate the central processing unit. Upon receipt of an input, the "processors" follow the processing procedures, interrogate and update files as required, and produce outputs. This step does not operate in microseconds, but it does do an excellent job of detecting errors in logic.

MANAGEMENT FACTORS

Although this discussion emphasizes analysis of operations, the general approach taken by management to information system development can do much either to negate good work or overcome inadequacies. These factors, then, certainly deserve mention.

A major contribution management can make is to establish a "developmental environment." The first aspect of this environment is an orientation to only system development. The goal is to complete a system that meets the design criteria, and the participants in the project are judged by this goal. The system should therefore have no operational requirements prior to final testing. Complex information systems have much in common with complex weapon systems, yet, the management approaches for the two differ drastically. If, for example, Cape Kennedy were charged with defending the country, the missile program wouldn't have a chance. Once backlogs and operations schedules appear, information system development stops; quick fixes and patches consume the time of all available personnel. In a developmental environment, furthermore, problems can receive realistic evaluation and attention. Serious problems are the natural result of trying to do something new.

A second aspect of a developmental environment is its view of a data-system project as a real-world laboratory experiment. A number of progressive organizations are spending millions on "simulation," yet they ignore the sources within reach for gaining very "real" knowledge. An organization does not learn much about data system development merely from going through the process. A few people may learn something, but people come and go. For the organization to profit, it requires a design and analysis group specifically charged with identifying controversial areas, collecting data, and recording results. In this way, we may begin to learn something more substantial about information system development.

A final aspect of a developmental environment is the need for special skills and training. To become a full-fledged engineer, a man requires at least four years of college and several on the job. At that point, he is allowed to develop simple pieces of hardware. But to develop a million- or billion-dollar information system, programmers get an average of four weeks training, and analysts often get none. If information system projects are to proceed efficiently, more and better training is required, and a first step is to devote more work on developing proper content.

Another major question facing management is design packaging. Most system people agree that the first step of a project is to define at a gross level the total system, the sub-packages, and the interrelationships among sub-packages. In an inventory system, factory distributions, due-in control, and purchase-order generation might each comprise a package. At this point, the controversy arises: should the packages and the total system become operational at a single date in the future, or should the available people be assigned to fully man the most important package or packages? When they finish this package, they would move on to the next. One might call these the parallel and the series approaches.

Intuition favors the series approach. Shortening the design phase for each package limits the effects of policy and personnel changes during development. The project is also easier to defend if something is running after only one year instead of five. The early packages will serve as test vehicles to check out controversial ideas. If they fail to work, they can be eliminated with a minimum of inconvenience. The training and techniques acquired in early packages help out in subsequent ones. Finally, concentrated development of one package gives the manager much better control potential than would the simultaneous development of many packages. In integrated systems, series packaging may require the manual performance of certain functions at very low efficiency. The situation

is only temporary, however, and manual performance of parts of the system may aid greatly in providing experience to improve the computerized version.

It is hard to acquire factual knowledge on series versus parallel design, as well as on most other design questions. Here is certainly an area where formal analysis could be most useful. These are the sorts of outputs that a real-world experiment approach will, perhaps, one day yield.

SECTION V

CONCLUDING REMARKS

Although the application of formal analysis to information system problems has been limited in the past, it is possible and useful to develop more formal structures for information system design. A substantial body of knowledge relevant to the area exists. Statistical decision theory provides a guide to the evaluation of information system effectiveness. AUTOSATE and similar approaches simplify and improve the analysis of information flows and offer a starting point for costing. Tabular structures for stating decision processes help to formalize programming.

Information system design and development are still largely intuitive, and can profit from a great deal more attention to formal techniques. For example, the selection of appropriate design packaging - series or parallel development of sub-systems - is a promising area for formal analysis. The current requirements for Command and Control systems re-emphasize the importance of improving our ability to design information systems.

It is interesting to note that the majority of technical people in the Command and Control field are specialists in hardware design, while the major problems lie in determining information requirements, selecting good decision rules, and developing systems to implement these information structures and decision rules. This discrepancy may well be the most significant problem in the field.

REFERENCES

1. Barnard, Chester. Functions of the Executive, Harvard University Press, Cambridge, Massachusetts, 1938.
2. Simon, Herbert. Administrative Behavior, Macmillan, New York, 1946.
3. Bavelas, Alex. "Communication in Task-Oriented Groups," in H. Lasswell and D. Lerner (eds.), Policy Sciences, Stanford University Press, Stanford, California, 1951; H. Guetzkow and H. Simon, "The Impact of Certain Communication Nets Upon Organization and Performance in Task-Oriented Groups," Management Science, Vol. 4, 1955, pp. 380-402.
4. Shannon and Weaver, The Mathematical Theory of Communication, University of Illinois Press, 1949.
5. Wiener, Norbert. Cybernetics, The Technology Press of M. I. T., and John Wiley and Sons, New York, 1948; Ashby, W. R. Design for a Brain, John Wiley and Sons, New York, 1952; Forrester, J. W. Industrial Dynamics, John Wiley and Sons, New York, 1961.
6. Marschak, J. "Elements for a Theory of Teams," Management Science, Vol. 2, 1955, pp. 127-137; idem, "Toward an Economic Theory of Organization and Information," in Thrall, Coombs, and Davis (eds.), Decision Processes, John Wiley and Sons, New York, 1954, Chap. XIV, pp. 187-219.
7. Gregory, R. H., and Van Horn, R. L. Automatic Data Processing Systems, Wadsworth Publishing Company, San Francisco, 1961.
8. Gatto, O. T., McGuire, C. B. and Van Horn, R. L. Research Aspects of Command and Control, The RAND Corporation, RM-3155-PR, June 1962.
9. Savage, L. J. Foundations of Statistics, John Wiley and Sons, New York 1954.
10. Matthes, Theodore K. Missile Firing Procedures, The RAND Corporation, R. M. -3233, August, 1962.
11. Nelson, H. W. and Petersen, J. W. Integrated Supply Support Policies: The LP-III Experience, The RAND Corporation, RM-2839-PR, January, 1962.

12. Gatto, O. T. AUTOSATE: An Automated Data Systems Analysis Technique, RM-3118-PR, The RAND Corporation, May, 1962.
13. Pollack, Solomon L. DETAB-X: An Improved Business-Oriented Computer Language, The RAND Corporation, RM-3273-PR, August, 1962.
14. Van Horn, R. L. Systematic Methods for Programming Simplification, The RAND Corporation, P-2447, September 1961.

**OPERATIONS RESEARCH INFORMATION TECHNOLOGY
AND INFORMATION SYSTEMS**

OPERATIONS RESEARCH, INFORMATION TECHNOLOGY, AND INFORMATION SYSTEMS

C. A. Wogrin* and D. F. Votaw, Jr. **

SECTION I

INTRODUCTION

The development of information systems is based on a new and complex technology -- "information technology." Although this technology has made immense strides, the connections between it and more traditional areas of knowledge are not yet well understood. Moreover, it seems reasonable to hope that a better understanding of these connections will lead to important advances in the state-of-the-art of development of information systems.

The purpose of this paper is twofold. First, to describe fundamental characteristics of information systems. Second, to point out important relationships between operations research and information technology.

It will emerge from the discussion that: a) operations research can contribute to the structuring of information systems; b) perhaps the most significant contribution of operations research to information-system development can be made at the interface in an organization between the command (management) and the information system used by the command.

*Yale University, New Haven, Connecticut

**The MITRE Corporation, Bedford, Massachusetts

SECTION II

INFORMATION SYSTEMS

To talk about systems in any meaningful way requires some definitions and explanations for the purpose of selecting from a vast collection of things (generally thought of as systems) a particular class to which we intend to address our remarks. The dictionary definition of the word "system," meaning an assemblage of objects united by some form of regular interaction or interdependence, is a good starting point. We further restrict attention to what Hall ^[1] has called an "open system." An open system is characterized by its existence in a world external to the system, responsive to specific attributes of the external world, and providing outputs to which the external world is responsive. In this sense, a system may be viewed by the external world as an object embedded in a system, and any system, in turn, may contain objects which in themselves can be viewed as systems, that is, they can be subsystems of systems. It is not necessary that a subsystem be of the same class as the system, but it is possible, in many cases, to find subsystems that are of the same class. In a system science, it would be necessary that the subsystem be of the same class as the system or that there be well-defined relations between classes of systems. These are observations which will not be pursued further in this paper.

We will further restrict ourselves to systems which have a useful purpose in the organized society of men. In other words, we are thinking of systems incorporated in business and manufacturing, military command and control, and the like. This is not a definition of a class of systems but is the motivating concept for any definitions which follow.

From the external world, the system can be viewed as illustrated in Fig. 1. In this instance, the system is seen as an object having inputs and outputs. The inputs and outputs are divided into three classes: energy, materials and data. Data for this purpose can be both numeric and symbolic. Numeric data are facts about the external world represented by a number system amenable to arithmetic operations, while symbolic data are facts represented in a form not necessarily intended to be manipulated by the rules of arithmetic. A particular system may not have inputs and outputs from all three classes, and any input is, in general, a vector quantity.

Before attempting the analysis or synthesis of the systems of interest here, it seems quite reasonable to require that the purpose of the system be clearly understood and the inputs and outputs well defined. In the case of synthesis (or design) of a system, it may be necessary to defer the specification of the inputs to later stages of the design process, but, in the end, a well-designed system should not operate on unknown inputs with undefined methods. (This is an objective to be reached with the evolution of a system science rather than a supportable present-day fact.)

A final observation about the inputs and outputs is that the system has no control over the source of the inputs from the external world nor over any of the effects of the outputs on the external world. To argue otherwise merely extends the system to include a larger number of components.

A system as described above and represented in Fig. 1 is, in general, a collection of machines, men, procedures, communication networks and controls. Because of the varied aspects of this list, it is unlikely that there can be any single system science that can formally discuss all of the problems of design and analysis of the system. We will, therefore, refer to Fig. 1 as the "ground

From the external world, the system can be viewed as illustrated in Fig. 1. In this instance, the system is seen as an object having inputs and outputs. The inputs and outputs are divided into three classes: energy, materials and data. Data for this purpose can be both numeric and symbolic. Numeric data are facts about the external world represented by a number system amenable to arithmetic operations, while symbolic data are facts represented in a form not necessarily intended to be manipulated by the rules of arithmetic. A particular system may not have inputs and outputs from all three classes, and any input is, in general, a vector quantity.

Before attempting the analysis or synthesis of the systems of interest here, it seems quite reasonable to require that the purpose of the system be clearly understood and the inputs and outputs well defined. In the case of synthesis (or design) of a system, it may be necessary to defer the specification of the inputs to later stages of the design process, but, in the end, a well-designed system should not operate on unknown inputs with undefined methods. (This is an objective to be reached with the evolution of a system science rather than a supportable present-day fact.)

A final observation about the inputs and outputs is that the system has no control over the source of the inputs from the external world nor over any of the effects of the outputs on the external world. To argue otherwise merely extends the system to include a larger number of components.

A system as described above and represented in Fig. 1 is, in general, a collection of machines, men, procedures, communication networks and controls. Because of the varied aspects of this list, it is unlikely that there can be any single system science that can formally discuss all of the problems of design and analysis of the system. We will, therefore, refer to Fig. 1 as the "ground

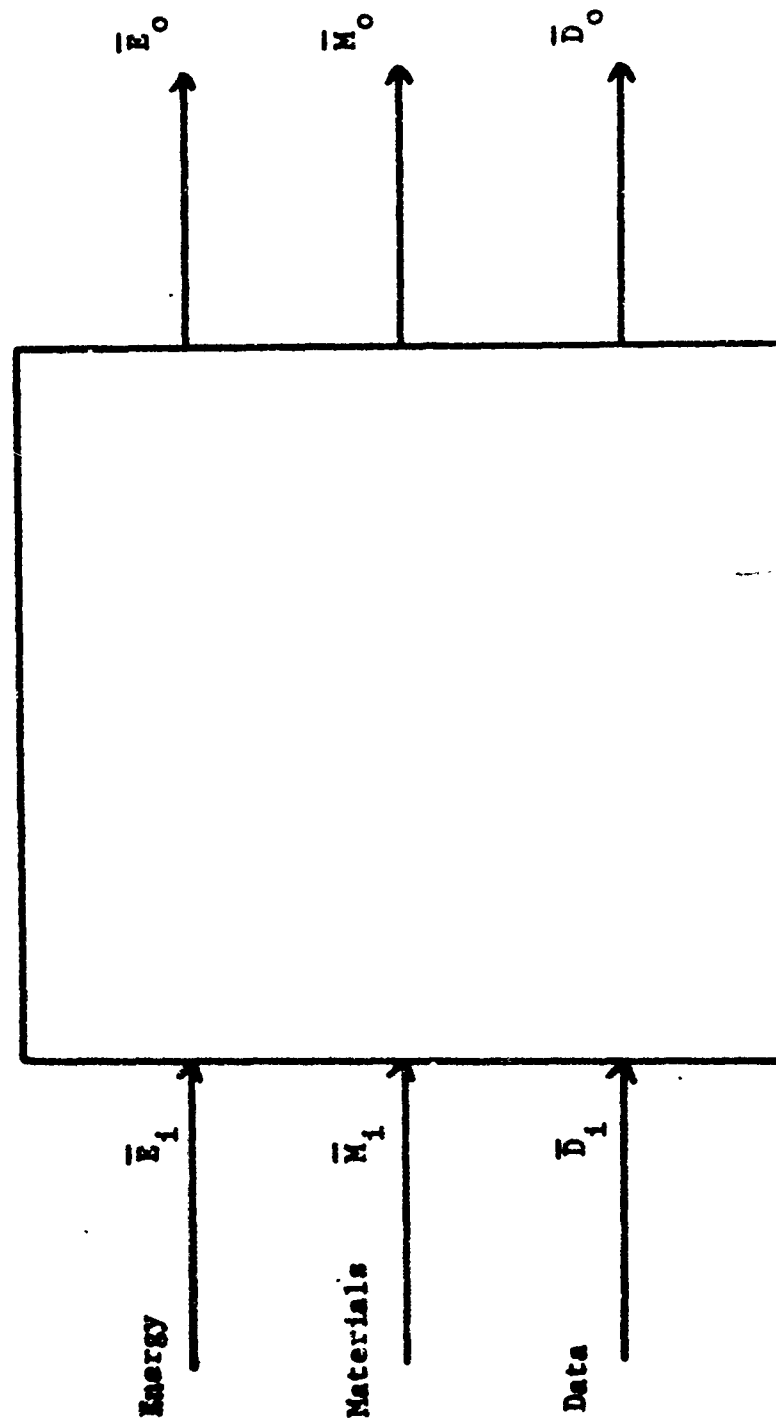


Fig. 1. The System Viewed as an Object

system" and not attempt any rigorous treatment of its internal structure in a single formal method. One could, for example, direct attention to a control system or to a communication system.

Figure 2 shows a first view of an internal structuring of the ground system. In this case, the attempt is not to find subsystems of the same class as the system (although this may be possible and in some cases useful), but to separate the system into types of operations.

The material and energy system concerns itself with the transporting, conversion, storing, and combining of energy and materials. No further consideration will be taken other than to note that measurements as to the status of this system are sent as data to the information system, and control of the processes is accomplished by means of the data provided by the information system. These are indicated by the lines labeled \bar{D}_1 and \bar{D}_2 in Fig. 2. There is no handling or manipulation of data in the energy and materials system.

The areas of particular interest in this session are those designated in Fig. 2 by the information system and the command (which could also be called the management or decision-maker). This portion of the system deals only with data. The division into information system and command is based upon operation and procedure for which: a) effective routines, which we will call algorithms, exist, and for which no algorithm exists. The former is in the information system and the latter in the command.

By this dichotomization, the information system is by definition algorithmic in its structure. It should, therefore, be amenable to analysis and one could be optimistic in talking about an information system science. In talking about the objects of the information system and the command, we include functions performed by humans or machines, but do not think of a human or machine as an object. Attention is focused on the function to be performed. Thus, we say the

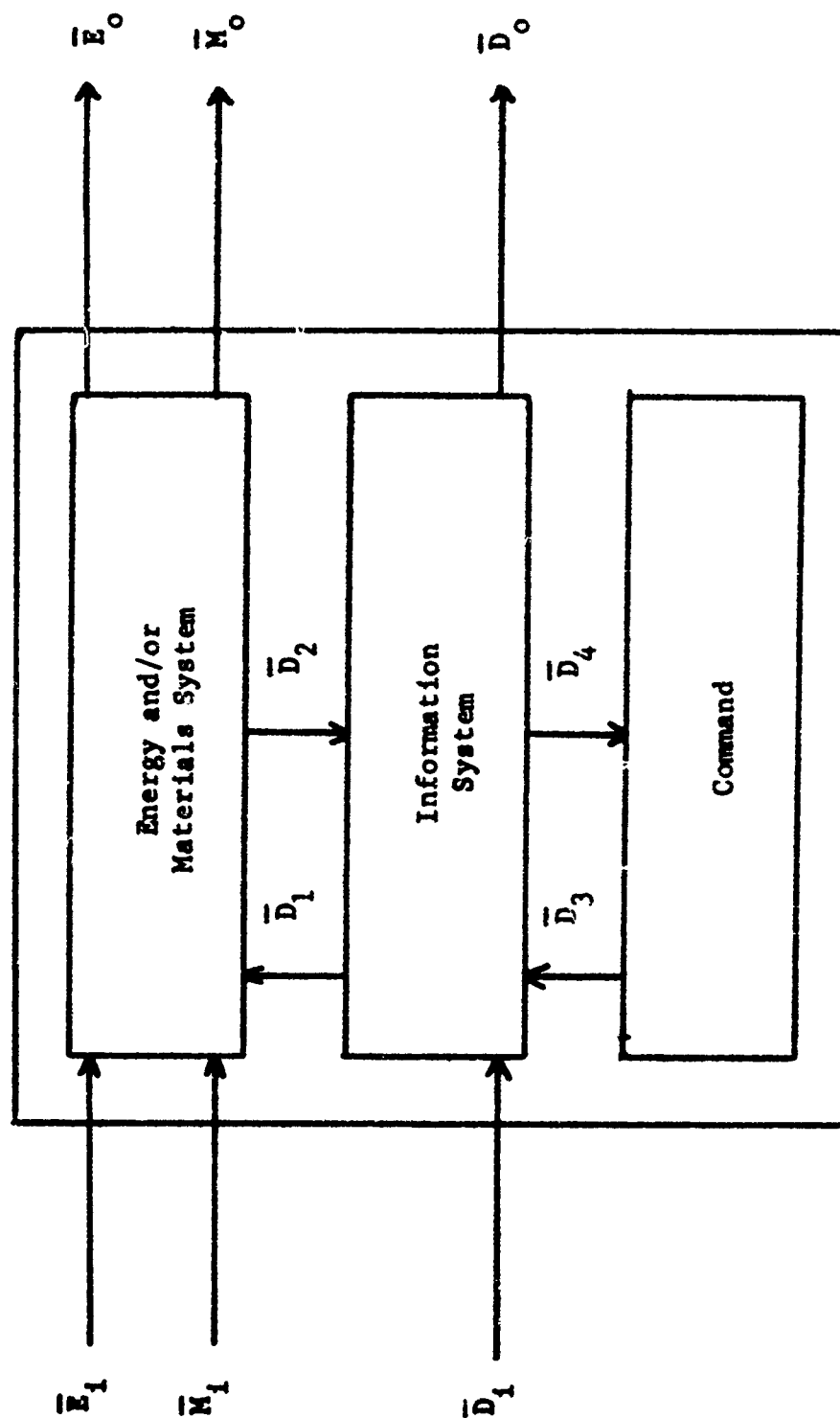


Fig. 2. The Partitioning of the System

command and not the commander. Some of the operations performed by the commander may be algorithmic. These are included in the information system. The command constitutes only those functions which are not algorithmic. Note further that the command has contact only with the information system. He receives data on \bar{D}_4 and sends data on \bar{D}_3 .

Attention will now be directed to the information system. The information system is essentially the data handling system with the understanding that data can be more than mere numbers. The following definitions are proposed:

Datum. A numeric or symbolic representation of a state or condition of some pertinent aspect of the external world or of a state or condition of an internal part of the ground system.

Function. An algorithm for operating on data. A function is a rule for combining, changing, or generating data.

Information System. A system whose objects are functions and in which interdependence of the objects is expressed by means of sequences of these functions. A sequence of functions can itself be a function.

An information system which has no provision for data storage is limited, at any instant in time, to present outputs that are dependent on and only on the inputs at that time. The more interesting and complex systems are those wherein a means of storage exists. Such systems can provide output data constructed from a history of the system and its environment.

The functions of an information system are not easily listed in any general way. The work of Turing, and those works stemming from his, show that there exists some elemental list of functions from which all algorithms can be constructed; but to describe a large system from a small list of functions would require very long sequences. Unnecessarily long sequences have limited

use just because of their complexity, and because a long sequence of functions to describe a single operation of a computer in a particular system would be an unnecessary complication. It would seem, therefore, that the list of specific functions is not a general problem but an individual problem.

For a science of information systems, therefore, the functions should be treated as classes with attention focussed on the properties of the classes, the relationship of the elements within a class, and the relationship between the classes. A list of classes can be as follows:

- (a) storage and retrieval,
- (b) calculation (arithmetic),
- (c) manipulation (non-arithmetic),
- (d) collection (input and internal measurement),
- (e) dissemination (output and internal routing), and
- (f) coding.

Sequences of functions for performing the desired operations are commonly represented by the flow diagrams used by computer programmers, designers, and others. An example of this is given in both of the other papers of this session and in many other papers. What is still lacking is a general method for analyzing the sequences, generating equivalent sequences, and optimizing among a set of equivalent sequences and theorems or methods for testing the adequacy of a sequence.

As a science of information systems develops, the classes, relationships within the classes, and relationship between classes will be rigorously defined. Since, however, an information system is not a pure abstraction, but exists as part of a ground system, the science cannot develop without careful attention to how the data within the system represent the pertinent aspects of the external

world and the ground system. The functions must be constrained in such a way that the operations upon data produce data that continue to be representative of the external world pertinent to the purposes of the system.

To satisfy this condition, then, it is necessary to devote attention in a specific system to the inputs and outputs and to the command. There must be assurance that the data supplied to the commander be complete and in an optimally useful form. The commander, in his role in the command, must know how to send data to the information system and, more importantly, must understand fully the extent of his control over the system.

Even though a true information system science does not exist, large information systems have been built (and they work), are in the process of development, and will be developed in the future. Success is due to a considerable art that has grown up through experience. The process requires many people representing, collectively, a wide variety of talents.

SECTION III

OPERATIONS RESEARCH AND INFORMATION TECHNOLOGY

OPERATIONS RESEARCH

As an organized activity, operations research began in England shortly before World War II. One of the earliest operations research groups assisted the RAF in setting up the early warning radar system. During World War II, operations research became an established activity in the British and American military organizations, and since that time it has continued to play an important role at top echelons in those organizations. For example, at present the Joint Chiefs of Staff have an operations research group known as the Weapons Systems Evaluation Group (WSEG).

The operations research discussed above is called military operations research, since military problems form the area of application. Some remarks are now in order regarding industrial operations research. Scientific management, which began in the late nineteenth century, is a forerunner of industrial operations research; furthermore, time-and-motion study, industrial quality control, and industrial engineering are early forms of industrial operations research. Since World War II, operations research groups have been set up in many industrial firms in the United States -- perhaps because of the brilliant success of military operations research during the war.

Thus far in the discussion, no definition of operations research has been given. More than ten years ago, the following definition was popular: "operations research is a scientific method of providing executive departments with a quantitative basis for decisions regarding the operations under their control^[2] ." This definition was quite appropriate at the time of publications, and it expresses an important aspect of present-day operations research. It should be added,

however, that in the last decade the subject has broadened and deepened to such an extent that a comprehensive definition is now almost impossible.^[3] There is an extensive overlap among operations research, management science, systems analysis, and systems engineering. Furthermore, there is some reason to believe that from these and related areas of activity a new area is emerging which might appropriately be called system science.*^[1, 3,]

INFORMATION TECHNOLOGY

This area can be described in various ways. Simon^[4] has indicated that it might be regarded as analogous to the "automation of mental work." Schultz and Whisler^[4] assert that this technology "... is a means of organizing information, of relating it to various managerial decision problems, and, in many instances, of working out decisions based on predetermined and programmed rules." According to them, this technology involves three kinds of activities (all pertaining to the quantitative analysis of management problems): a) use of mathematical and statistical methods; b) use of computers for mass data processing, and c) application of computer-based simulation to decision-making.

The hardware and software associated with information technology includes communication equipment, sensors, display equipment, accounting machines, computers, and computer programs.

Information technology contains the resources for the development of information systems. One of the most important problems facing this technology today is that of advancing the state-of-the-art of information-system development.^[5, 6, 7, 8]

*A stable terminology has not yet developed. To include all current variants, we might use the expression system(s) science(s).

OPERATIONS RESEARCH AND INFORMATICS SYSTEMS

Operations research is strikingly relevant to information-system development. In both these areas management decision-making is of central importance. Furthermore, some of the major activities underlying information-system development are also involved in operations research. (For example, activities of the kinds listed in the first paragraph of the preceding subsection have long been standard activities in operations research work.) In view of the urgent need for advancing the state-of-the-art of information system development and in view of the vast investment now being made in large-scale information systems for command and control, it would not be surprising if operations research plays an outstanding role in the development of future information systems.

Some of the ways in which operations research can contribute to information-system development are discussed below, in rather general terms. [4, 9, 10, 11]

Using the framework presented in section II, we can point out two basic problems on which operations research work is needed. The first concerns the interface between the information system and the command; one is defined as algorithmic, and the other as non-algorithmic. If the information system is totally automated, then it is indeed algorithmic. An important question that must then be dealt with is: are there any operations of the command which can be reduced to algorithms and thus incorporated into the information system? The second concerns the structuring of the information and the representation of the external world within the framework of the languages and codes of the information system.

The length of time involved in the process of determining requirements for a system and then planning, designing, and acquiring the system may be several years; furthermore, the number of man-years involved may be quite large. The fundamental considerations that must be dealt with in the course of

information-system development include such matters as reliability, performance, cost, resource-allocation, and scheduling. In many cases, such questions can be handled only by means of modelling or simulation. [5, 6]

The analysis of information requirements and the analysis of information flows are two major problems in information system development. Van Horn [12] has proposed some interesting ways of dealing with these problems. His approach involves the use of AUTOSATE (Automated System Analysis Technique.) One noteworthy feature of AUTOSATE is that it offers a starting point for costing a system.

The preceding discussion deals with various ways in which operations research can contribute to information systems. Some comments are now in order regarding a complementary type of contribution, namely, that in which information systems contribute to operations research. This type is illustrated by the fact that in many operations research studies programmed high-speed computers are used to carry out the extensive calculations involved. (It should be remarked here that a programmed computer is a realization of some of the functions of an information system.)

Donaldson and Harrison [13] describe a very interesting use of an information system in connection with a war game called THEATERSPIEL. (War games are important tools in certain kinds of operations research studies.) The information system employed provides not only high-speed calculation but also information storage and retrieval. Use of the information system has resulted in a considerable increase in speed of play. The same paper describes the use of a computer in connection with the U. S. Army War College war games. The feasibility of using a remotely located digital computer was demonstrated, and it was found that the effectiveness of the war games was improved.

REFERENCES

1. Hall, Arthur D. A Methodology for Systems Engineering, D. Van Nostrand Company, Inc., Princeton, New Jersey, 1962.
2. Kimball, George E. and Morsé, Phillip M. Methods of Operations Research, John Wiley and Sons, New York, 1951.
3. Flood, Merrill M. "New Operations Research Potentials," Operations Research, Vol. 10, No. 4 (1962), pp. 423-436.
4. Shultz, George P. and Whisler, Thomas L. (eds.) Management Organization and the Computer (Proceedings of a Seminar Sponsored by Graduate School of Business - University of Chicago and The McKinsey Foundation), The Free Press of Glencoe, Illinois, 1960.
5. Geisler, Murray A. "Integration of Modelling and Simulation in Organizational Studies" (pp. 139-148 in Management Sciences, Models and Techniques by Churchman, C. West and Michel Verhulst, Vol. 1 (Proceedings of the Sixth International Meeting of the Institute of Management Sciences held in Paris, 7-11 September 1959), Pergamon Press, New York, 1960).
6. Geisler, M. A. and Steger, W. A. "The Use of Manned Simulation in the Design of an Operational Control System," paper presented at The Joint IRE-AIEE-ACM Computer Conference, Los Angeles, California, May 9-11, 1961.
7. Henderson, Paul B. Jr. A Theory of Data Systems for Economic Decisions, Dewey Library, Massachusetts Institute of Technology, June 1960 (unpublished Ph. D. thesis).
8. "Mathematical Models for Information Systems Design and a Calculus of Operations," Magnavox Research Laboratories, ASTIA Document 255 677, 27 October 1961.
9. Gallagher, James D. Management Information Systems and the Computer, American Management Association, Inc., New York, 1961.
10. Jacobs, John F. "Air Force Command and Control System Development," SR-23, The MITRE Corporation, Bedford, Massachusetts, July 1961.
11. Malcolm, Donald G. and Rowe, Alan J. (eds.) Management Control Systems, John Wiley and Sons, New York, 1960.
12. Van Horn, R. L. "Formal Structures for Information System Design," ESD-TDR-62-474-11.
13. Donaldson, John L. and Harrison, Joseph O., Jr. "Information Systems for Man-Machine War Games," ESD-TDR-63-474-11.